Welcome to PK! This tutorial applies to the latest version, 2.x. For a tutorial for version 1.0, please go here.

There are 32 and 64 bit versions of PK. If all of the machines in your cluster are 64 bit, I recommend the 64 bit version. If any of the machines are 32 bit, you need to use the 32 bit version.

A PK cluster can be made by booting a computer using the CD, or by booting a virtual machine using the CD image. I recommend booting up the PK master node in a virtual machine. If you do so, you stay in your familiar operating system environment (Linux, Windows,...) and you don't need to burn any CDs. The compute nodes can still run real, full speed, unvirtualized Linux, regardless of whether or not the master node is virtual or real, so there is no loss of performance. PK versions are also tested on QEMU and VMware Server before release. These are both freely downloadable and work on both Linux and Windows. If you run PK using either of these systems, you should have no trouble. Other virtualization systems will most likely work, too, but I don't have time to test PK on all of them. To learn how to set up QEMU and VMware to run PK, see this section of this Tutorial. Setting up virtualization is a little work, but it's not hard, and it is a more productive way to use PK.

If you choose to boot your computer using the CD, it will most likely give you no trouble, but in some cases you will need to use cheatcodes to get it to work with your hardware.

Either way, you need to download the PK CD image. I recommend checking the md5 sum of your downloaded image with the correct sum posted on the download page to make sure that your image is not corrupted. If you decide to burn it to a CD, use a reasonably low speed.

A couple of notes on security: PK is now quite secure, except that whoever has access to the main node, either physically or across the net, can do anything they want to any of the computers in the cluster. Even though PK doesn't touch the hard drives of any of the machines in the cluster (except for using linux swap space, if it exists), hard disk partitions can be mounted. So don't let malicious or incompetent people use PK on your computers! The compute nodes are difficult to get into, except though the master node.

Networking etiquette and tips: The PK master node becomes a dhcp server, in order to assign IPs to the compute nodes. Furthermore, it is a very simple minded dhcp server - it will give an IP address to whatever computer asks for one. SO DON'T USE PK ON AN EXISTING NETWORK, or you'll get in trouble with the administrators, since PK's dhcp server might start giving out IPs to other people's computers, and people will not be able to get their work done. Another potential problem is that a PK node that boots up and receives an IP address that starts with 192.168.x.x automatically becomes a compute node. So if your intended master node is connected to a network that has a dhcp server that gives out those addresses, you'll have trouble (you'll arrive to a login screen with no way to enter). An easy solution is to pull the cables that connect the cluster to the rest of the world, turn off wireless cards, and then fire it up. Once your cluster is set up, you can shut off the dhcp server (`/etc/init.d/dhcp3-server stop`) and then connect to other networks.

Index of this tutorial

How does PK work?

If you're just in a hurry to get working, you can skip ahead, but eventually you may be curious to know more details about what is going on. This might help you to modify PK to make it better fit your needs.

When the frontend node boots:
When you first boot from CD, your hardware is detected and configured automatically. This process is similar to what occurs with all live CDs. Toward the end, the script `/cdrom/parallelknoppix.sh` does its work. This script determines whether or not the node is the frontend or a compute node, depending upon whether or not the string "192.168.0" is found in the output of the `ifconfig` command. Supposing it is not found, the node is considered to be the frontend. (This is why it is important to make sure that the frontend doesn't receive an IP address containing 192.168.0.x). A 200MB ramdisk is formatted and mounted at `/pkhome`. (The size of the ramdisk is determined by the contents of /cdrom/boot/isolinux/isolinux.cfg. Making the ramdisk larger requires remastering.) Then the knoppix user's home directory is decompressed from `/cdrom/pkhome.bz2` to `/pkhome/pkhome`. A symbolic link makes `/home/knoppix` point to `/pkhome/pkhome`. Then the node starts up the KDE desktop.

During the setup process:
The setup scripts in `/home/knoppix/PK_Setup/SetupScripts` are quite self explanatory, if you read them. Basically, networking is configured, ssh keys are generated, and the terminalserver script prepares the frontend node as a tftp boot server. `/pkhome/pkhome` is exported via NFS. The `RestartHPC` script integrates newly booted compute nodes into the cluster.

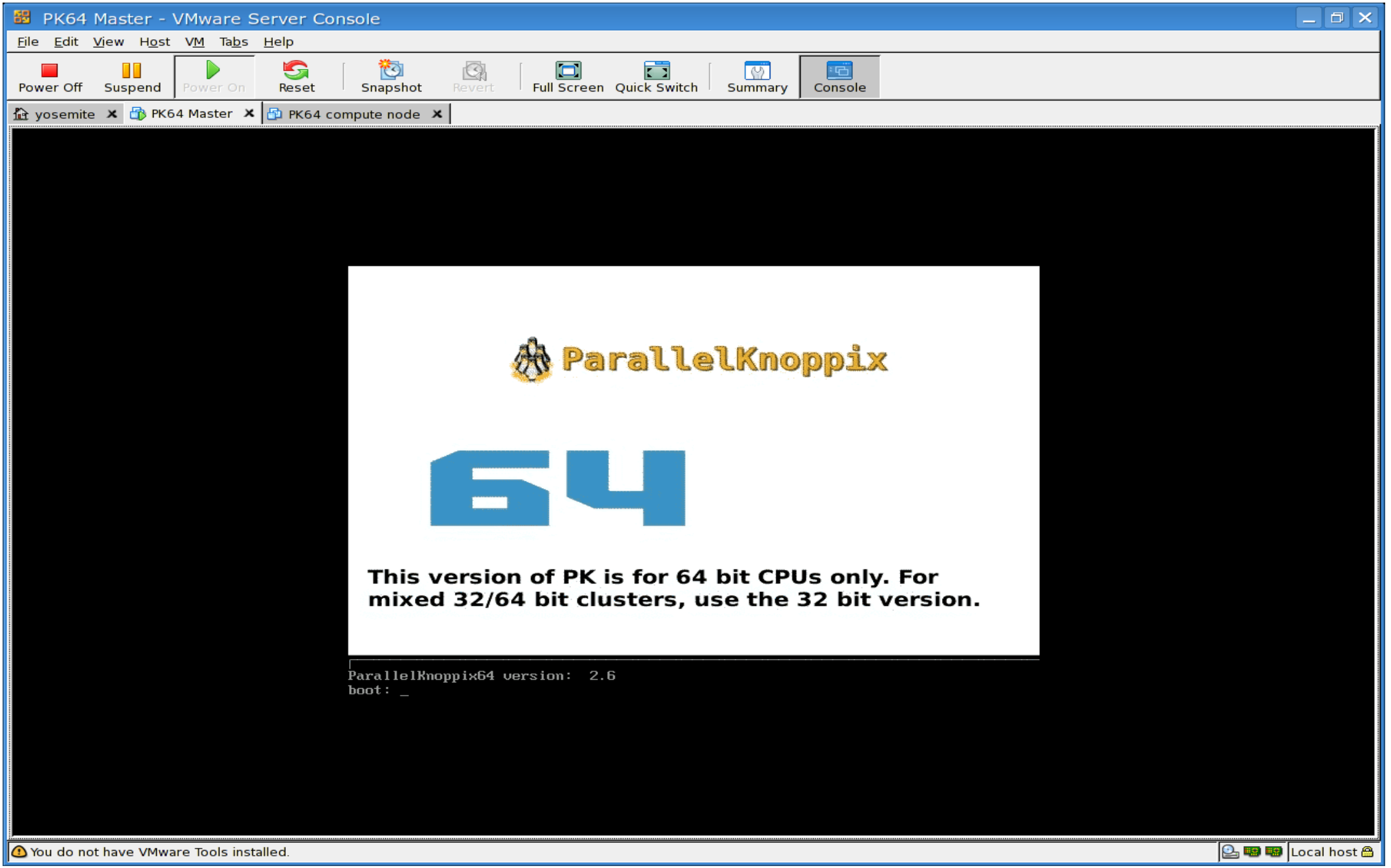
When a compute node boots:
When a compute node PXE boots, things work the same as with the frontend node, except that the filesystem is received via NFS from the frontend node. When the `parallelknoppix.sh` script runs, it determines that the node is a compute node. Then `/pkhome/pkhome` is NFS mounted from the frontend node, and linked as `/home/knoppix`. Finally, the `CallHome` script writes the node's IP address into a location on the shared home directory so that the frontend node will be aware that the compute node has booted up and can be integrated into the cluster.

When the RestartHPC script runs:
When the `RestartHPC` script runs on the frontend node, `/home/knoppix/tmp` is scanned, the compute nodes are found, and the file `/home/knoppix/tmp/bhosts` is written so that the MPI platforms are informed of the cluster setup. Note that you can modify that file by hand to customize your MPI environment. You might like to note which nodes are SMP, multi-core, etc. If the frontend node is running on a slow laptop or on a virtualization platform, you might like to exclude it from the computations, for example.
Back to index

Initial Setup

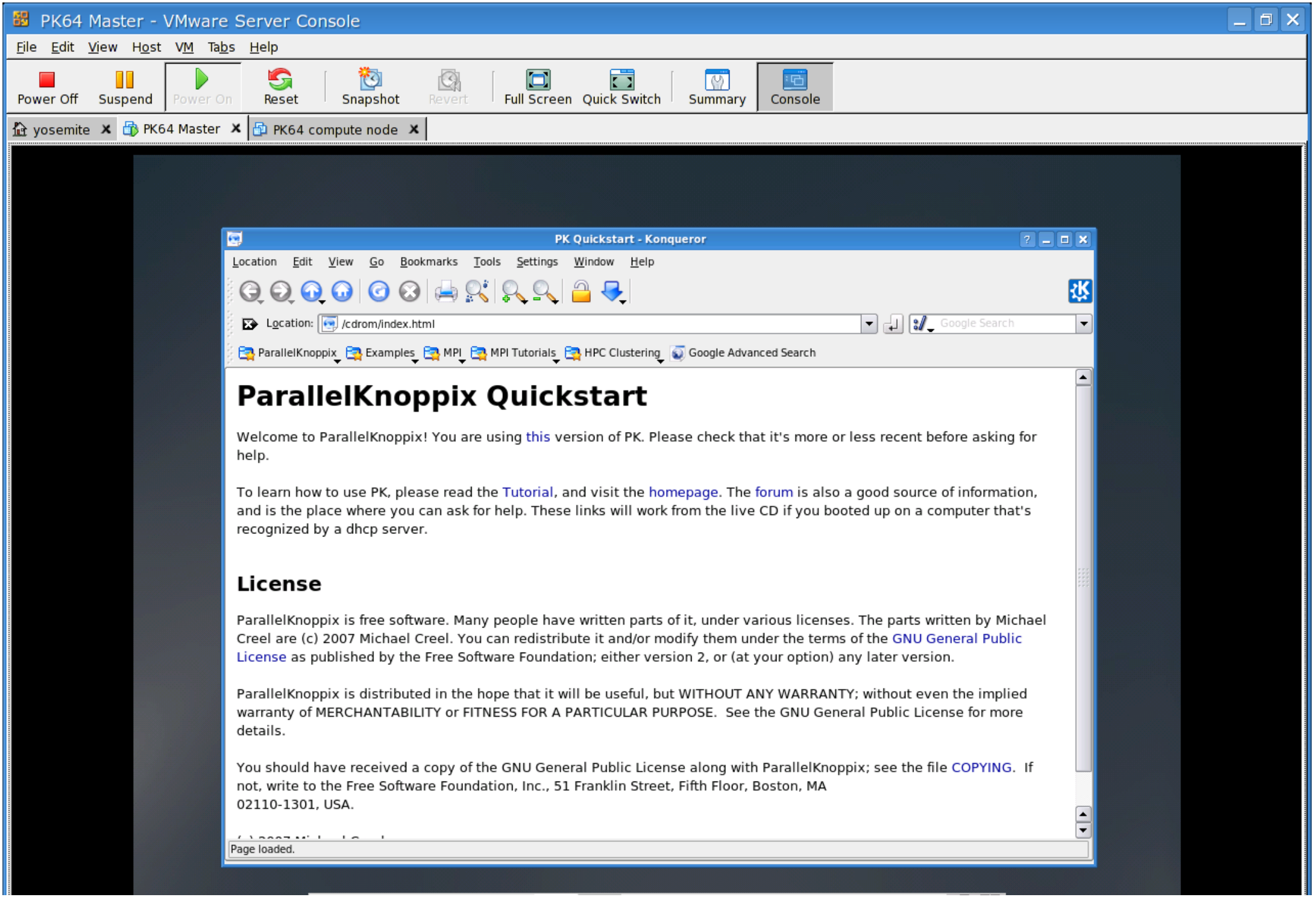
The screenshots here are big, so that you can see them well. **Scroll over to the R** in your browser to center them up.

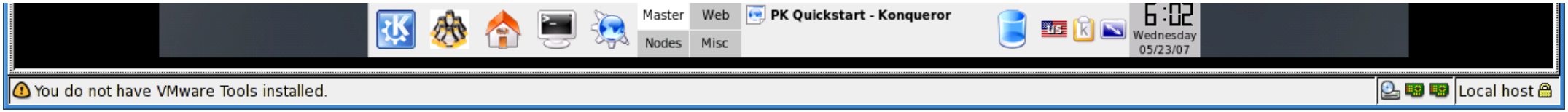
Boot your master node using the CD. You will see something like:



OK, first thing: slow down and read this before hitting enter. Note that the release version appears, right above the boot: prompt. Before continuing, you should make sure that there is not a newer release. You can hit F2 and F3 to get some information about boot options. There's more information on cheatcodes available on the Net, if you have trouble getting the master node to boot. OK, now you can hit enter....

Next you'll be at the initial screen inside the KDE desktop:


PK64 Master - VMware Server Console
File Edit View Host VM Tabs Help
Power Off Suspend Power On Reset Snapshot Revert Full Screen Quick Switch Summary Console
yosemite
PK64 Master
PK64 compute node
PK Quickstart - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: /cdrom/index.html
Google Search
ParallelKnoppix Examples MPI MPI Tutorials HPC Clustering Google Advanced Search
ParallelKnoppix Quickstart
Welcome to ParallelKnoppix! You are using this version of PK. Please check that it's more or less recent before asking for help.
To learn how to use PK, please read the Tutorial, and visit the homepage. The forum is also a good source of information, and is the place where you can ask for help. These links will work from the live CD if you booted up on a computer that's recognized by a dhcp server.
License
ParallelKnoppix is free software. Many people have written parts of it, under various licenses. The parts written by Michael Creel are (c) 2007 Michael Creel. You can redistribute it and/or modify them under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.
ParallelKnoppix is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
You should have received a copy of the GNU General Public License along with ParallelKnoppix; see the file COPYING. If not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
Page loaded.



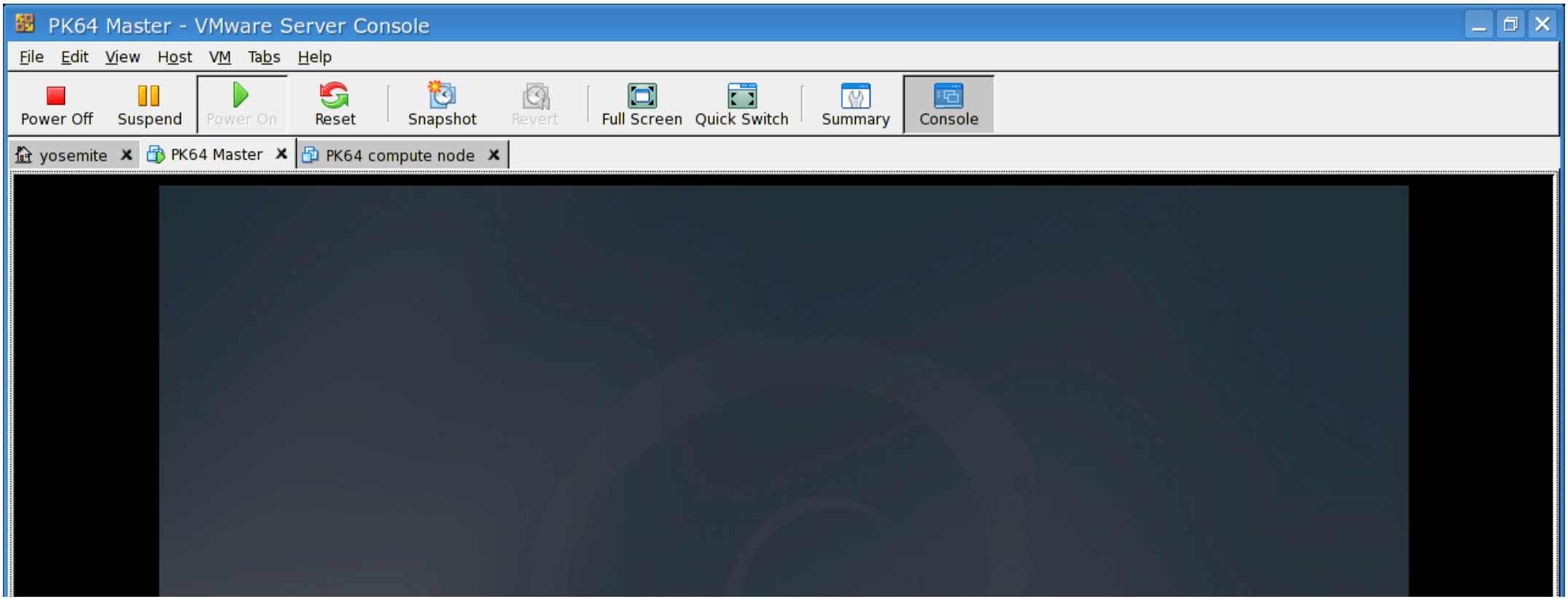
That screen points you to some links, including this tutorial. Please take the time to check the links out. They will work if you booted up on a computer that sees a dhcp server.

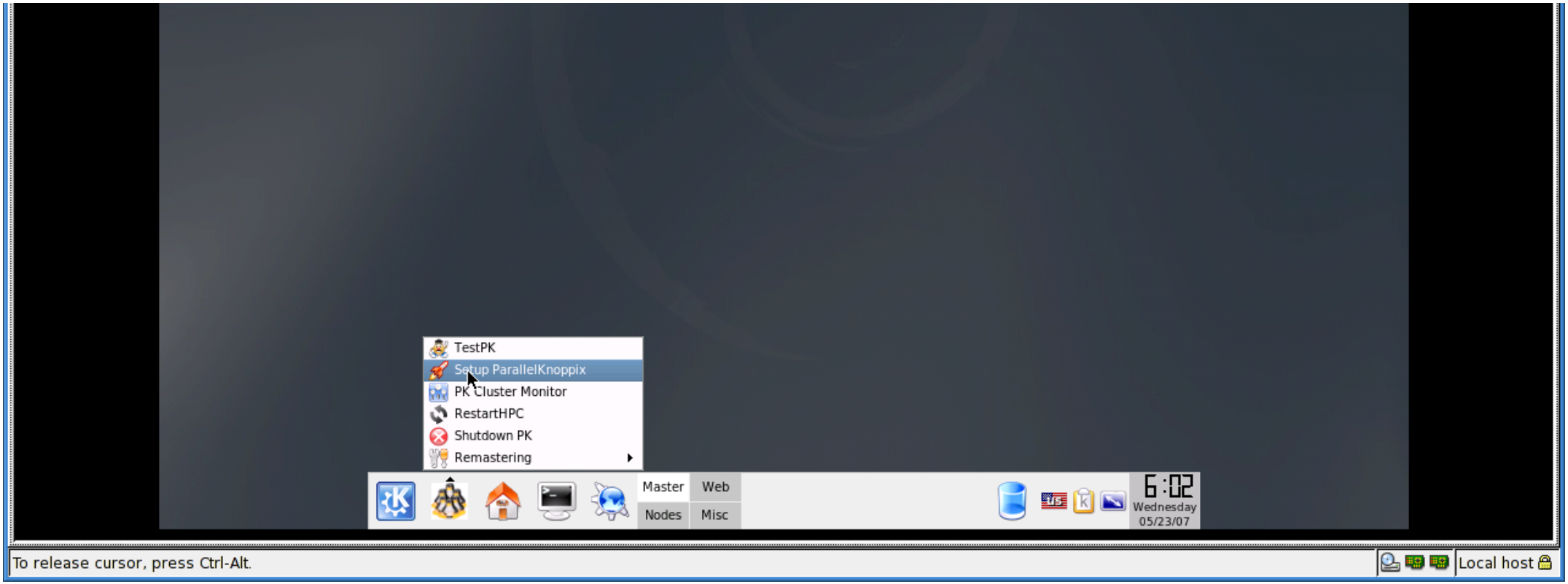
To set up a cluster, you need at least one more computer, or you can create a virtual cluster using a virtualization package like QEMU or VMware (which is what I'm using to make these screenshots). The computer you booted with the CD is the master node, and the other computers are the compute nodes. They need to be connected together in a network. You can use an existing ethernet, you can buy a switch and some cables, or to really keep it simple, you can use a crossover cable to connect a single compute node to the master node. I **recommend** disconnecting the master node from any network other than your cluster, at least until we take some steps to ensure that the external connection will be secure. This is also important to ensure that the compute nodes do not see any DHCP server other than the master node, which causes all kinds of headaches, both for you and for your co-workers (and then you again).

The compute nodes can be booted either using copies of the PK CD, or across the network, using the PXE boot capabilities of their network cards. This second option is **recommended**. You might need to learn a bit to make it work, but you'll be happy that you did it. If you just want to keep it extremely simple, use the CD method. You'll need a PK CD for each compute node. This works fine, provided your cluster is relatively small. It has the advantage that it works with network cards that don't do PXE boot.

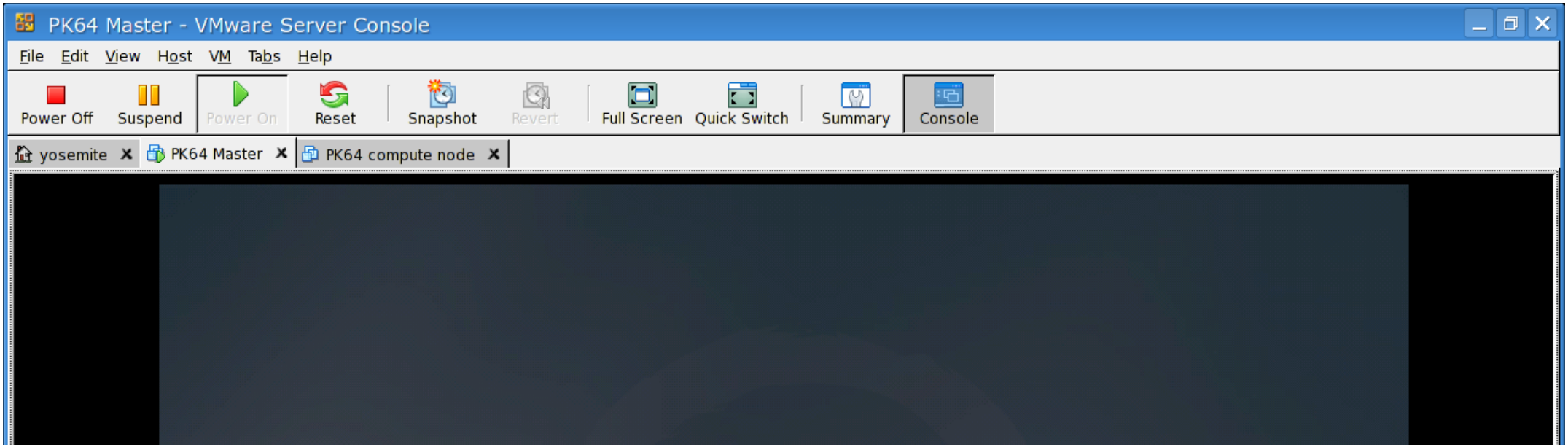
To use the PXE method, you may need to enable this feature in the BIOS setup routines of the compute nodes. Set the compute nodes to try PXE boot before booting from their hard drives. If your net cards are too old to do PXE boot, I recommend replacing them with newer ones, if you value your time at all. If you're unable to afford that, and you're willing to get into grimy details, rom-o-matic can be very useful.

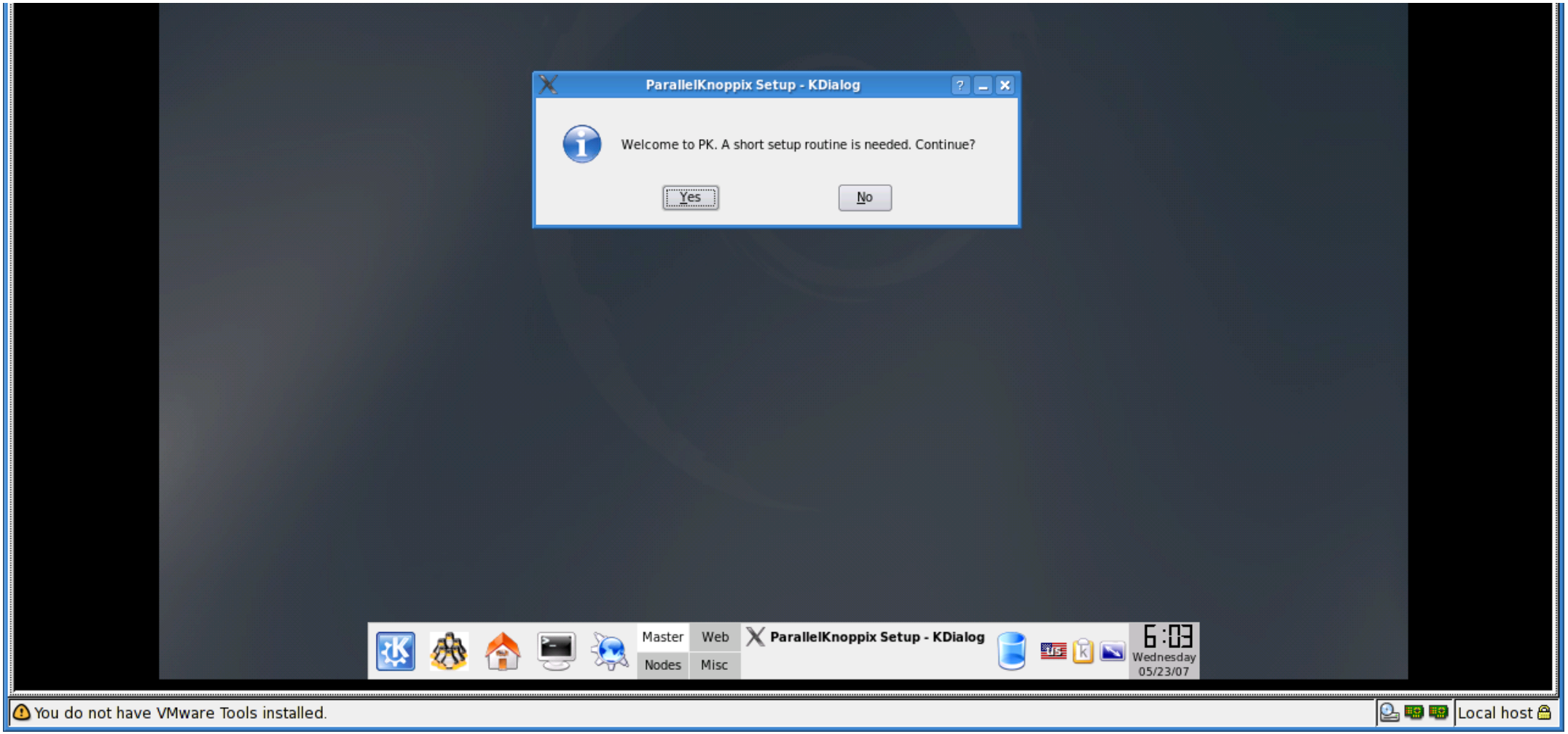
Assuming you have done the physical setup of a cluster, and the compute nodes are ready to net boot, we can get started. Find the **ParallelKnoppix** menu in the panel and click on the **Setup ParallelKnoppix** entry:



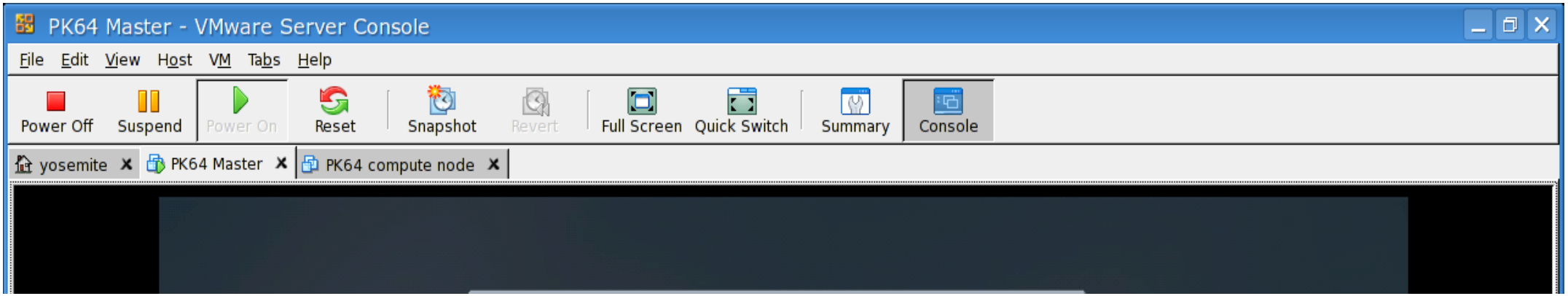


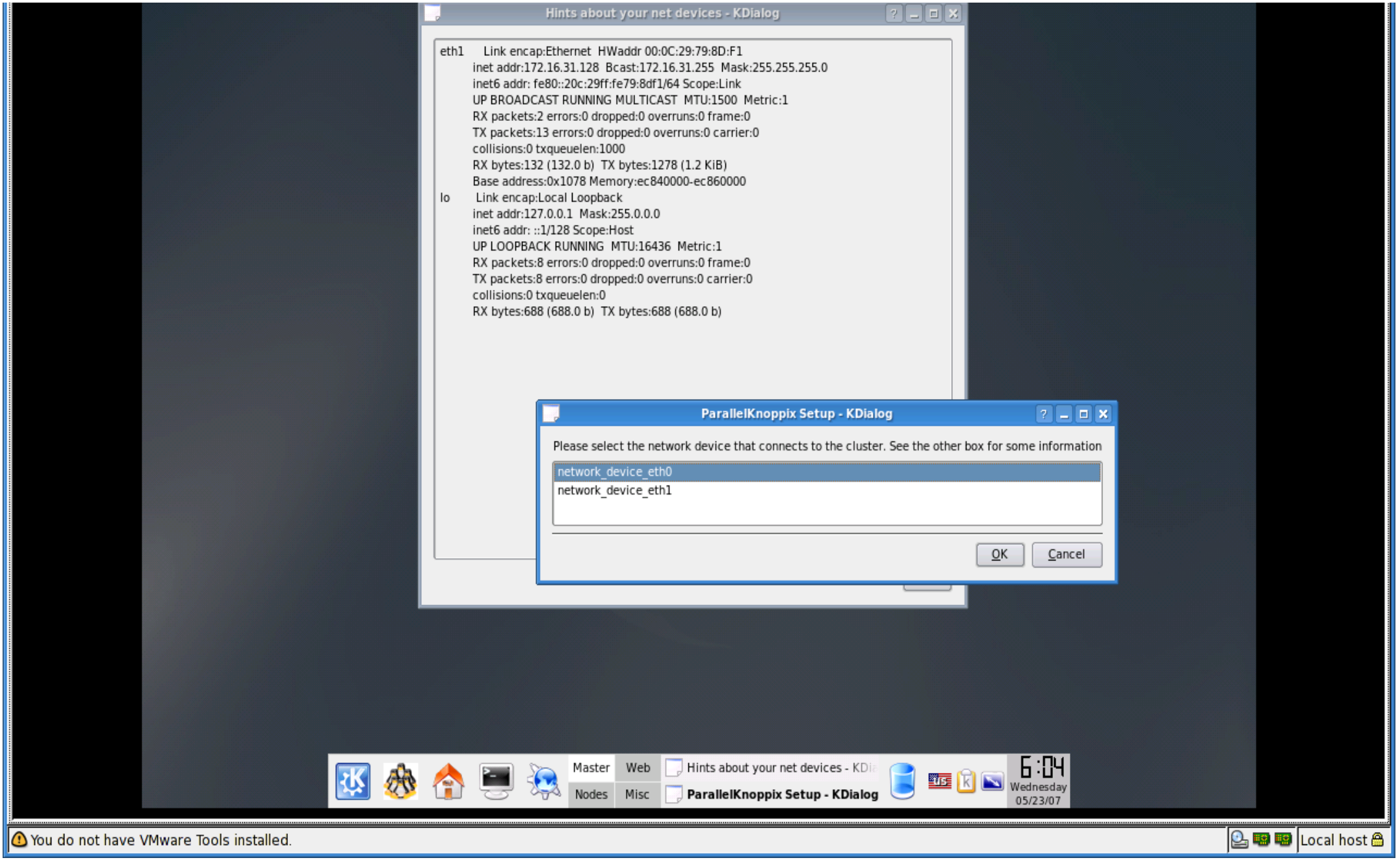
You see a little information:



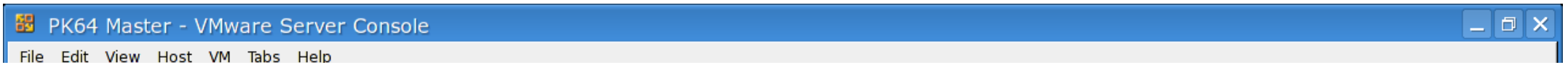


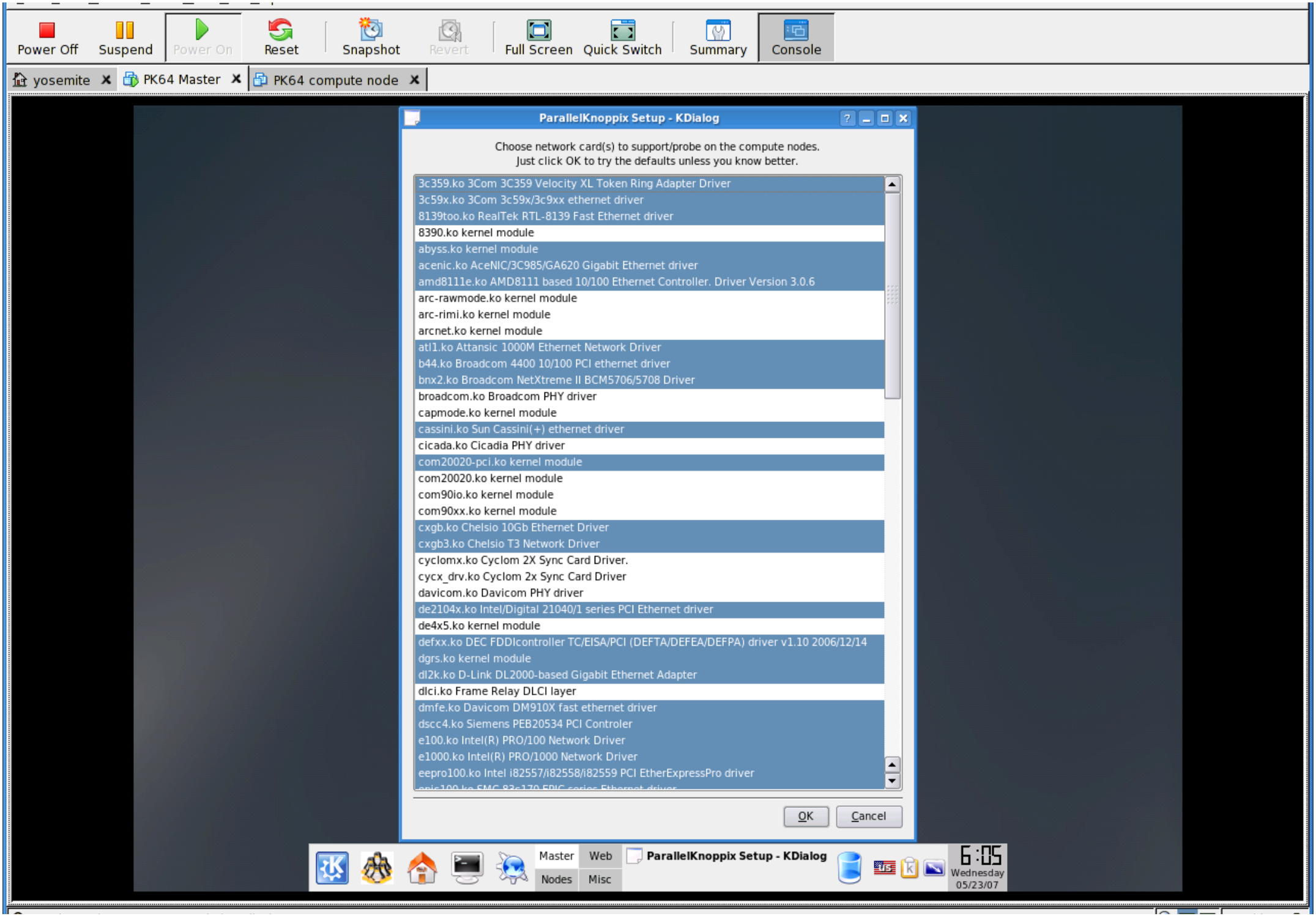
If you have only 1 networking device, the next screen will not appear. But if you're using the recommended virtualization setup, you will have 2 devices, so you'll see the following:





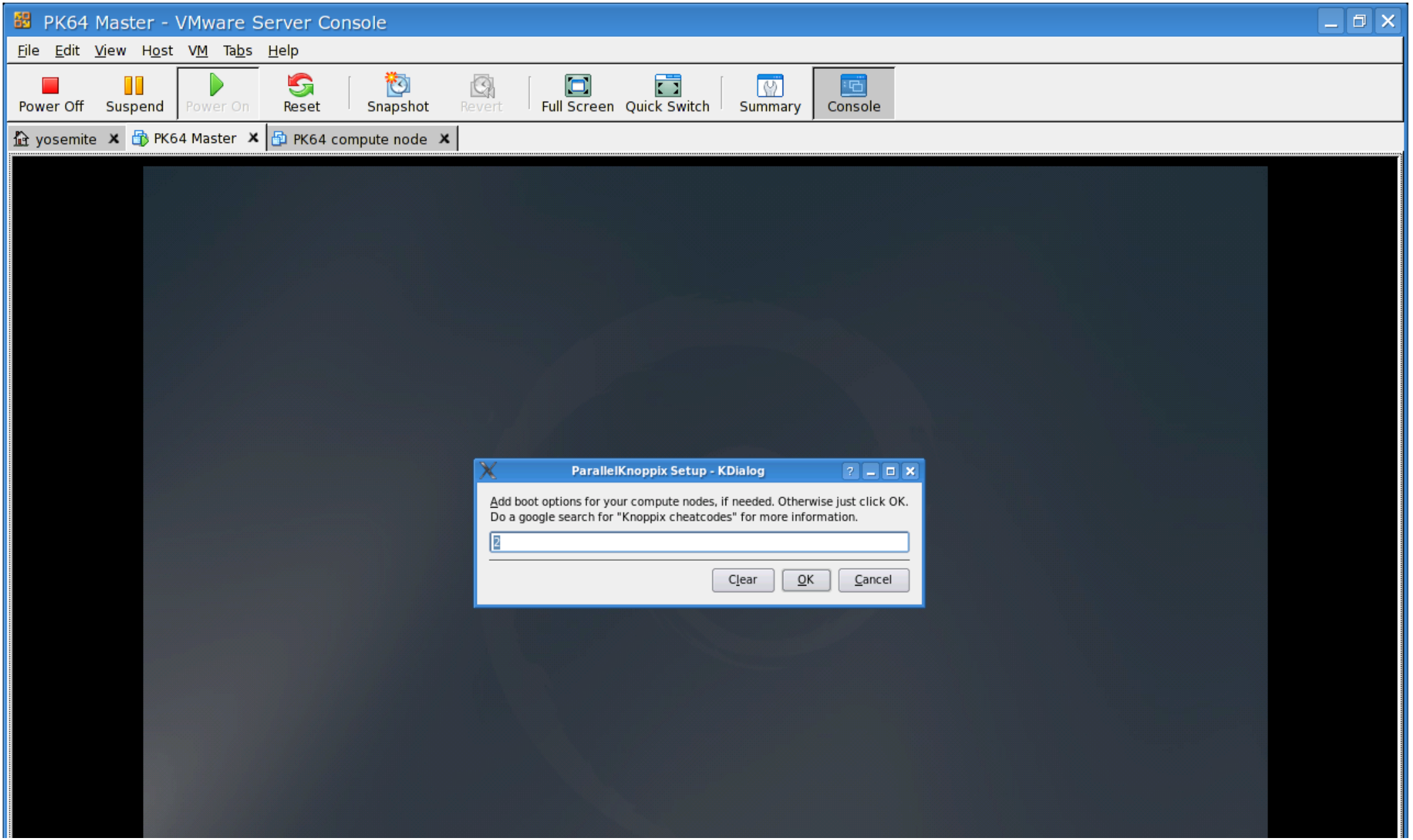
Select the device that goes to the cluster (in a virtualized machine, select the device that connects to the bridged real net device). On an aside, it's possible to set up a wireless cluster with PK, see this link. Next we see:

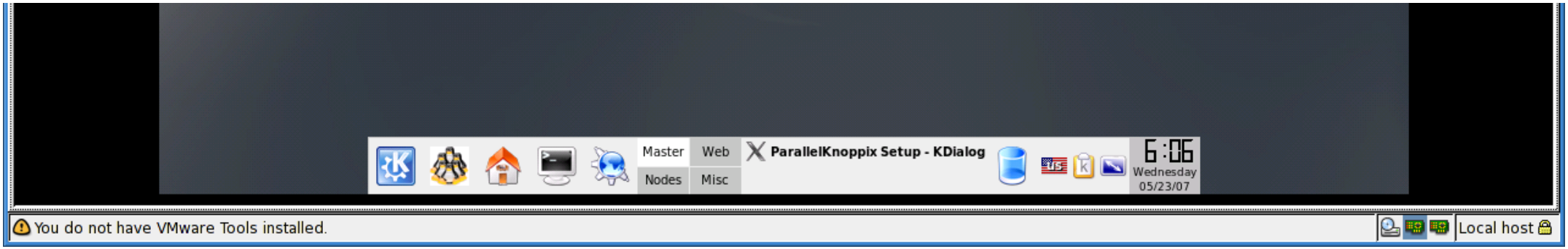



Power Off
Suspend
Power On
Reset
Snapshot
Revert
Full Screen
Quick Switch
Summary
Console
yosemite
PK64 Master
PK64 compute node
ParallelKnoppix Setup - KDialog
Choose network card(s) to support/probe on the compute nodes.
Just click OK to try the defaults unless you know better.
3c359.ko 3Com 3C359 Velocity XL Token Ring Adapter Driver
3c59x.ko 3Com 3c59x/3c9xx ethernet driver
8139too.ko RealTek RTL-8139 Fast Ethernet driver
8390.ko kernel module
abyss.ko kernel module
acenic.ko AceNIC/3C985/GA620 Gigabit Ethernet driver
amd8111e.ko AMD8111 based 10/100 Ethernet Controller. Driver Version 3.0.6
arc-rawmode.ko kernel module
arc-rimi.ko kernel module
arcnet.ko kernel module
atl1.ko Attansic 1000M Ethernet Network Driver
b44.ko Broadcom 4400 10/100 PCI ethernet driver
bnx2.ko Broadcom NetXtreme II BCM5706/5708 Driver
broadcom.ko Broadcom PHY driver
capmode.ko kernel module
cassini.ko Sun Cassini(+) ethernet driver
cicada.ko Cicadia PHY driver
com20020-pci.ko kernel module
com20020.ko kernel module
com90io.ko kernel module
com90xx.ko kernel module
cxgb.ko Chelsio 10Gb Ethernet Driver
cxgb3.ko Chelsio T3 Network Driver
cyclomx.ko Cyclom 2X Sync Card Driver.
cycx_drv.ko Cyclom 2x Sync Card Driver
davicom.ko Davicom PHY driver
de2104x.ko Intel/Digital 21040/1 series PCI Ethernet driver
de4x5.ko kernel module
defxx.ko DEC FDDIcontroller TC/EISA/PCI (DEFTA/DEFEA/DEFPA) driver v1.10 2006/12/14
dgrs.ko kernel module
dl2k.ko D-Link DL2000-based Gigabit Ethernet Adapter
dlci.ko Frame Relay DLCI layer
dmfe.ko Davicom DM910X fast ethernet driver
dscc4.ko Siemens PEB20534 PCI Controler
e100.ko Intel(R) PRO/100 Network Driver
e1000.ko Intel(R) PRO/1000 Network Driver
eepro100.ko Intel i82557/i82558/i82559 PCI EtherExpressPro driver
OK
Cancel
Master
Web
Nodes
Misc
ParallelKnoppix Setup - KDialog
6:05
Wednesday
05/23/07

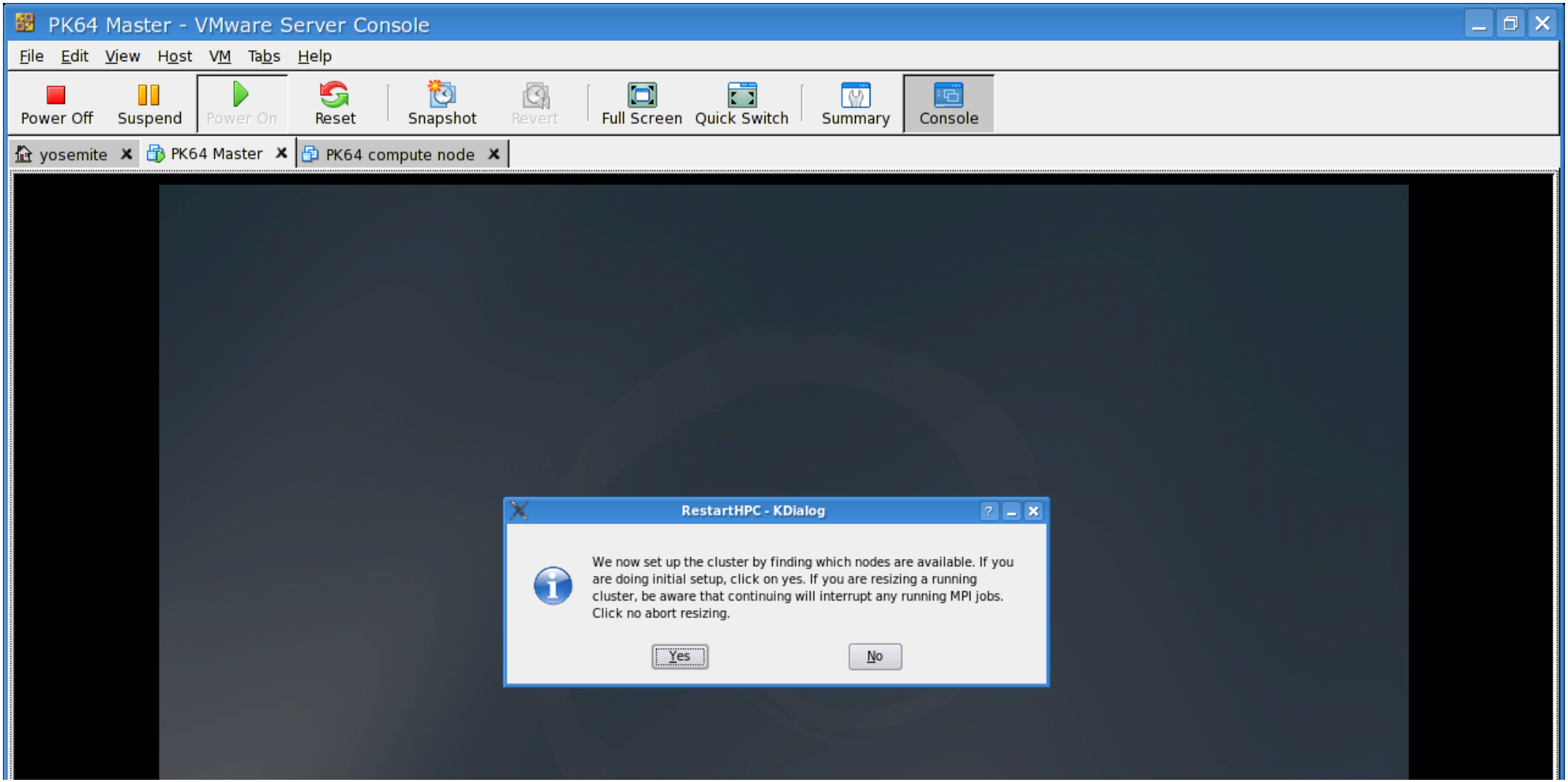
At this point, you need to include support for all the net devices your compute nodes use to attach to the cluster. With a bit of luck, the default selection will work. Just click OK unless you're quite sure that you shouldn't do so.

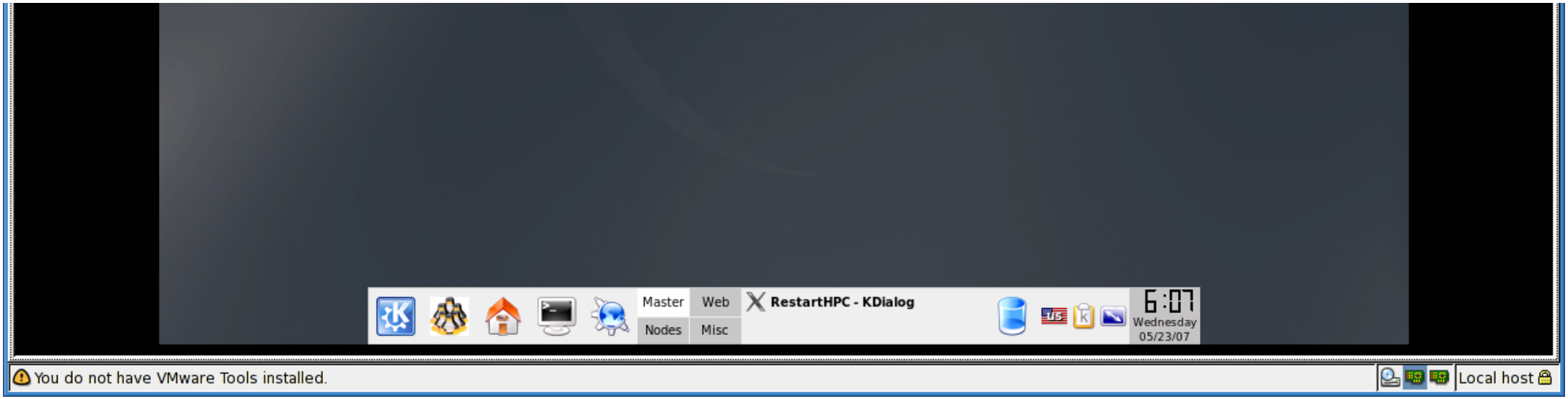
Next, you can add cheatcodes that you might need to make the compute nodes boot. Whatever you enter here is passed to all the compute nodes. Most people can just leave this the way it is. Some options to try if you don't have luck are noapic and acpi=off.



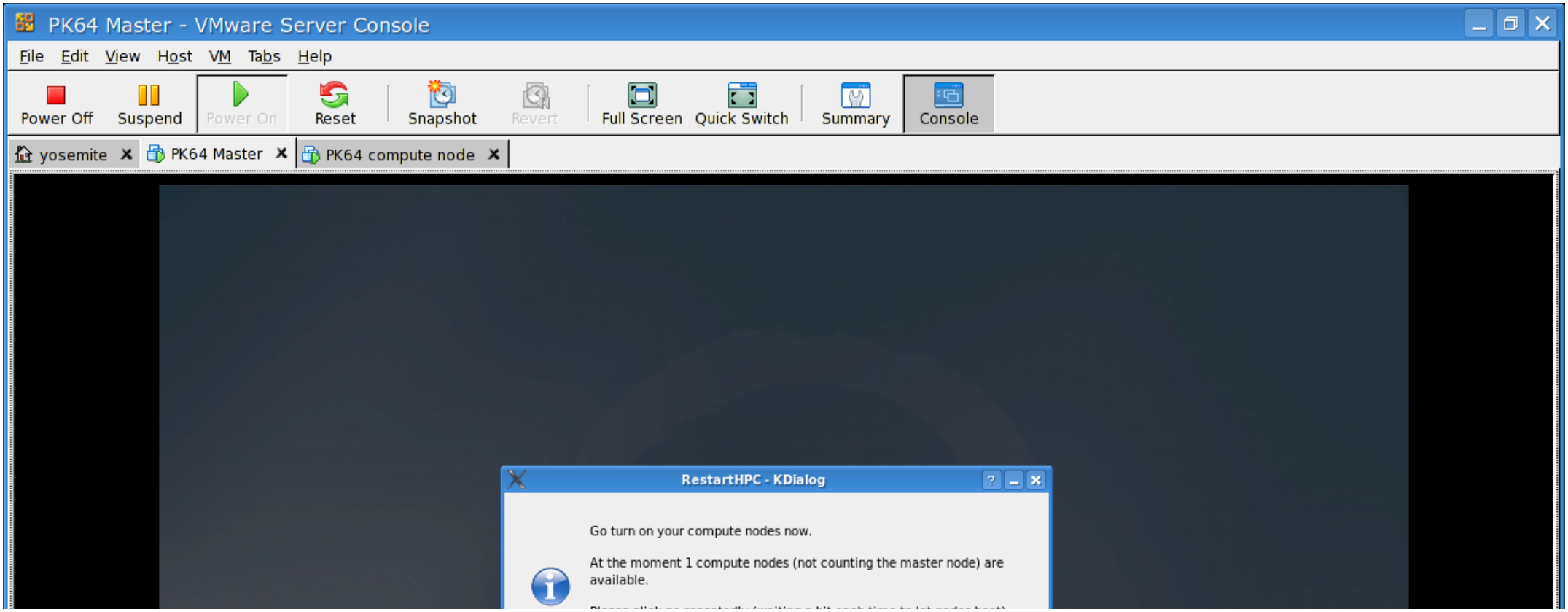


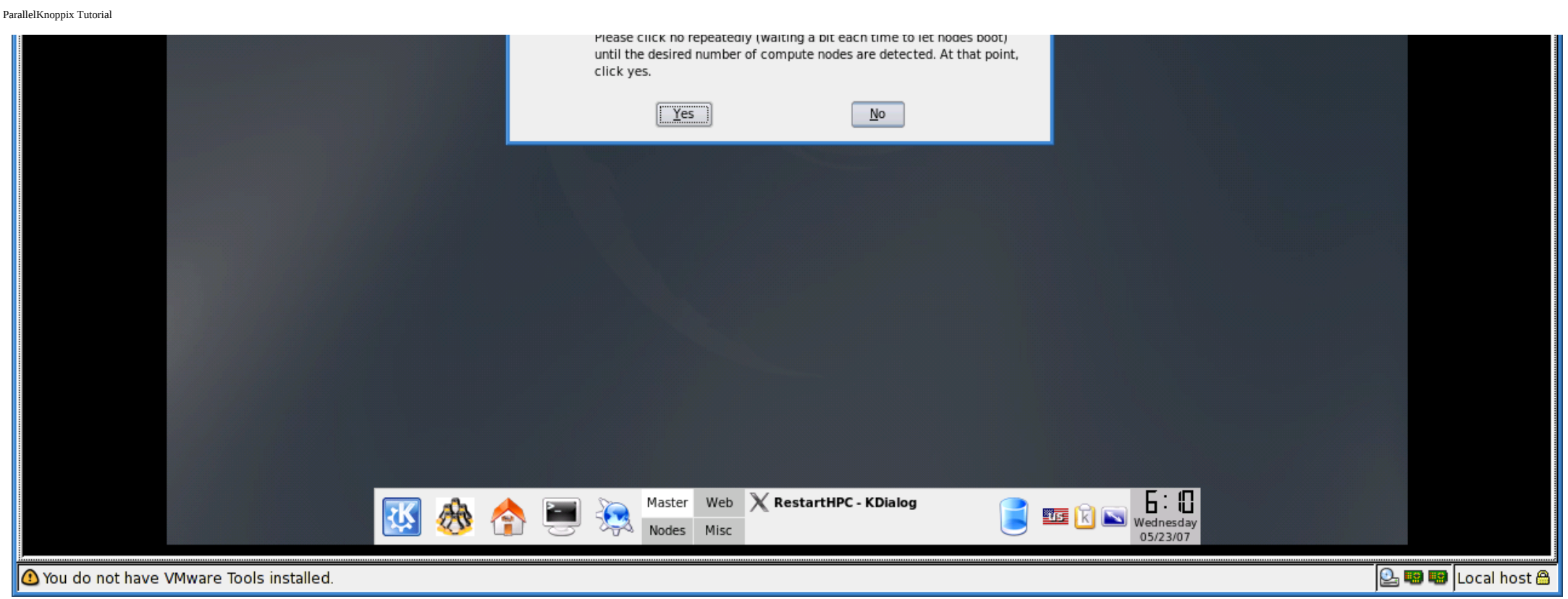
Now the cluster will be set up:



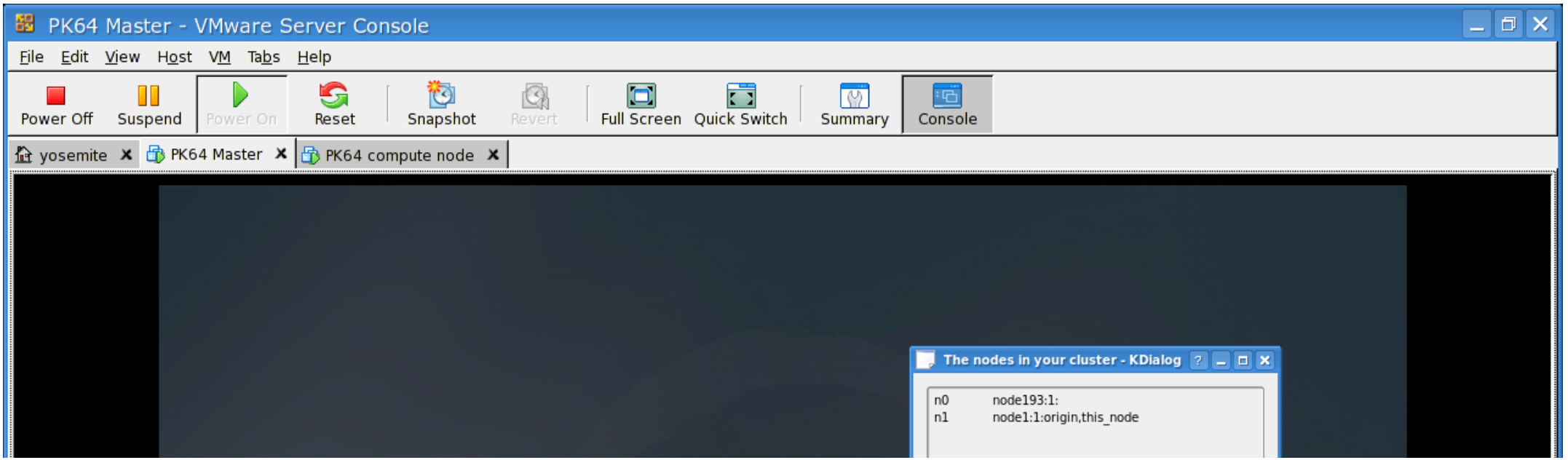


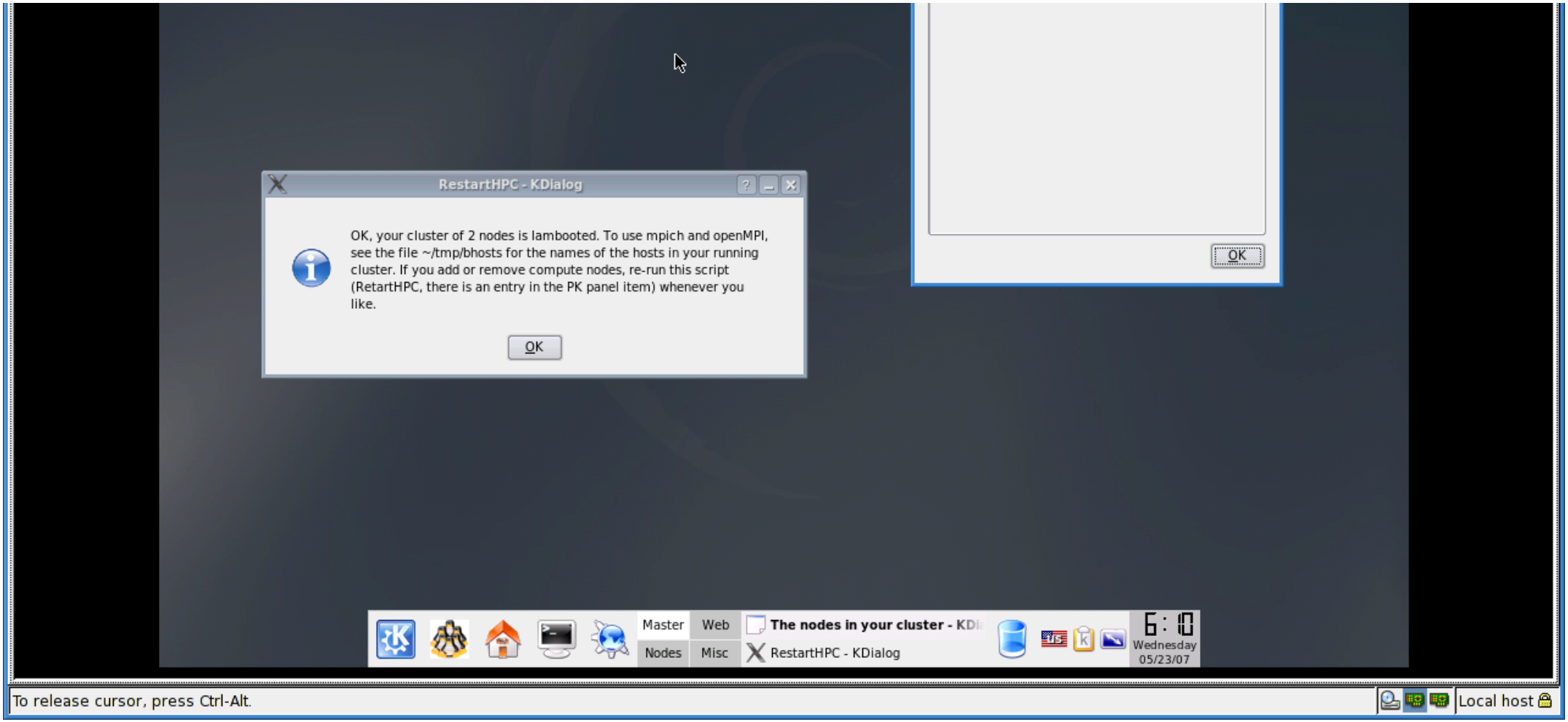
You're in a loop now, with the main node searching for compute nodes that have booted up. You should go turn on the nodes, and let them boot. Click on "no" to see how many nodes have been detected. When you have the desired number, click "yes". In the following shot, 1 compute node has come up:



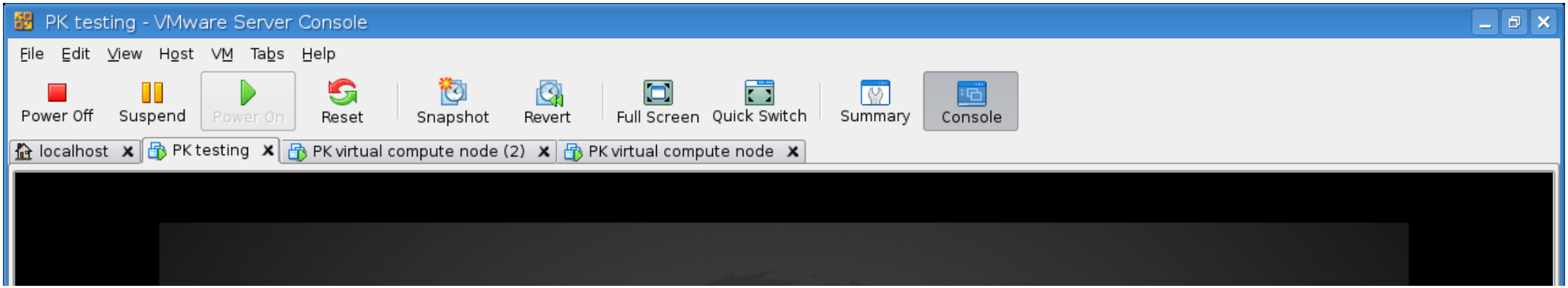


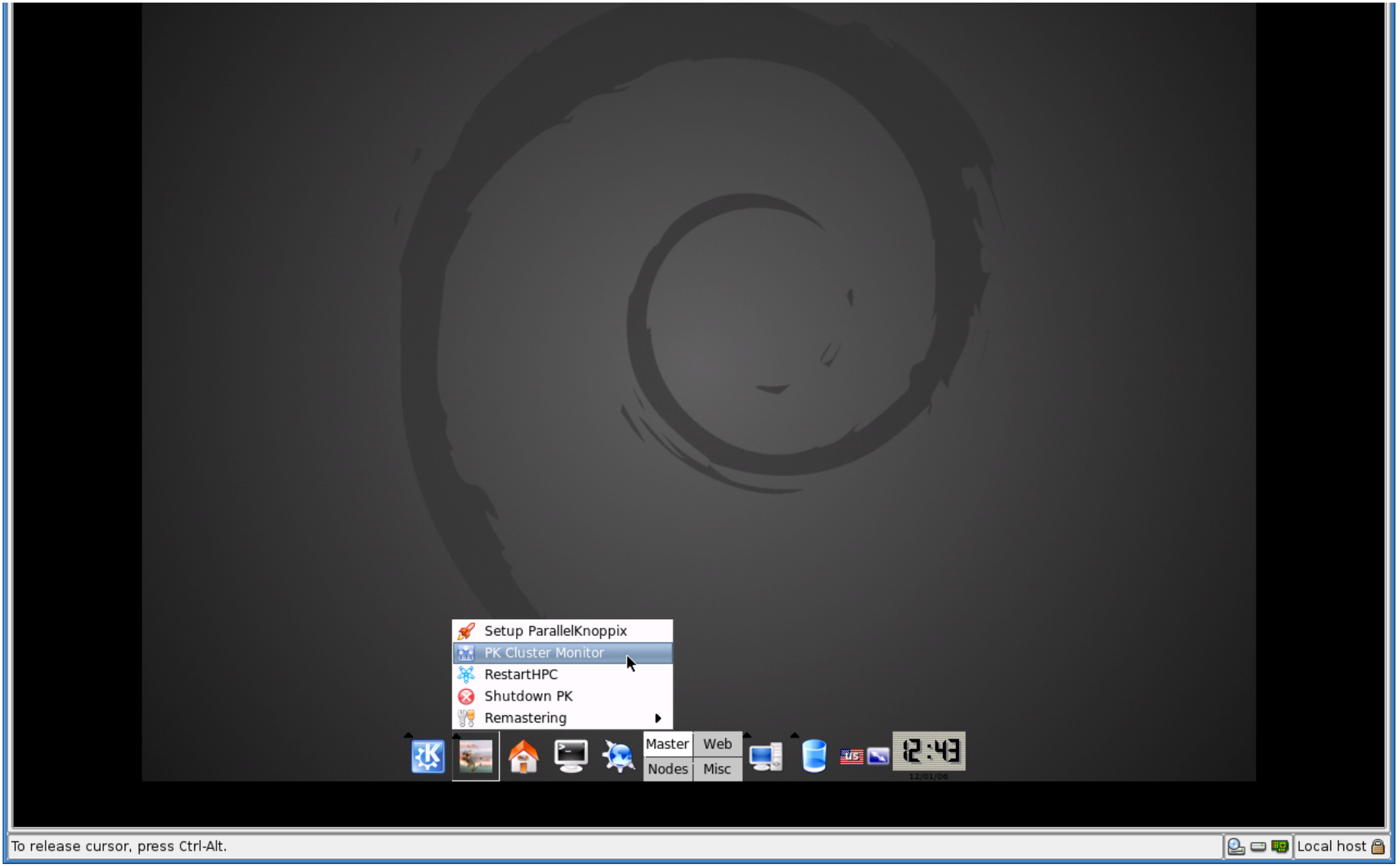
I click yes, and PK finishes setting up the cluster:



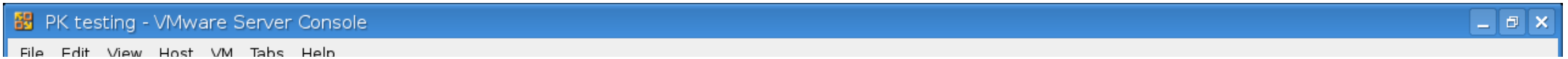


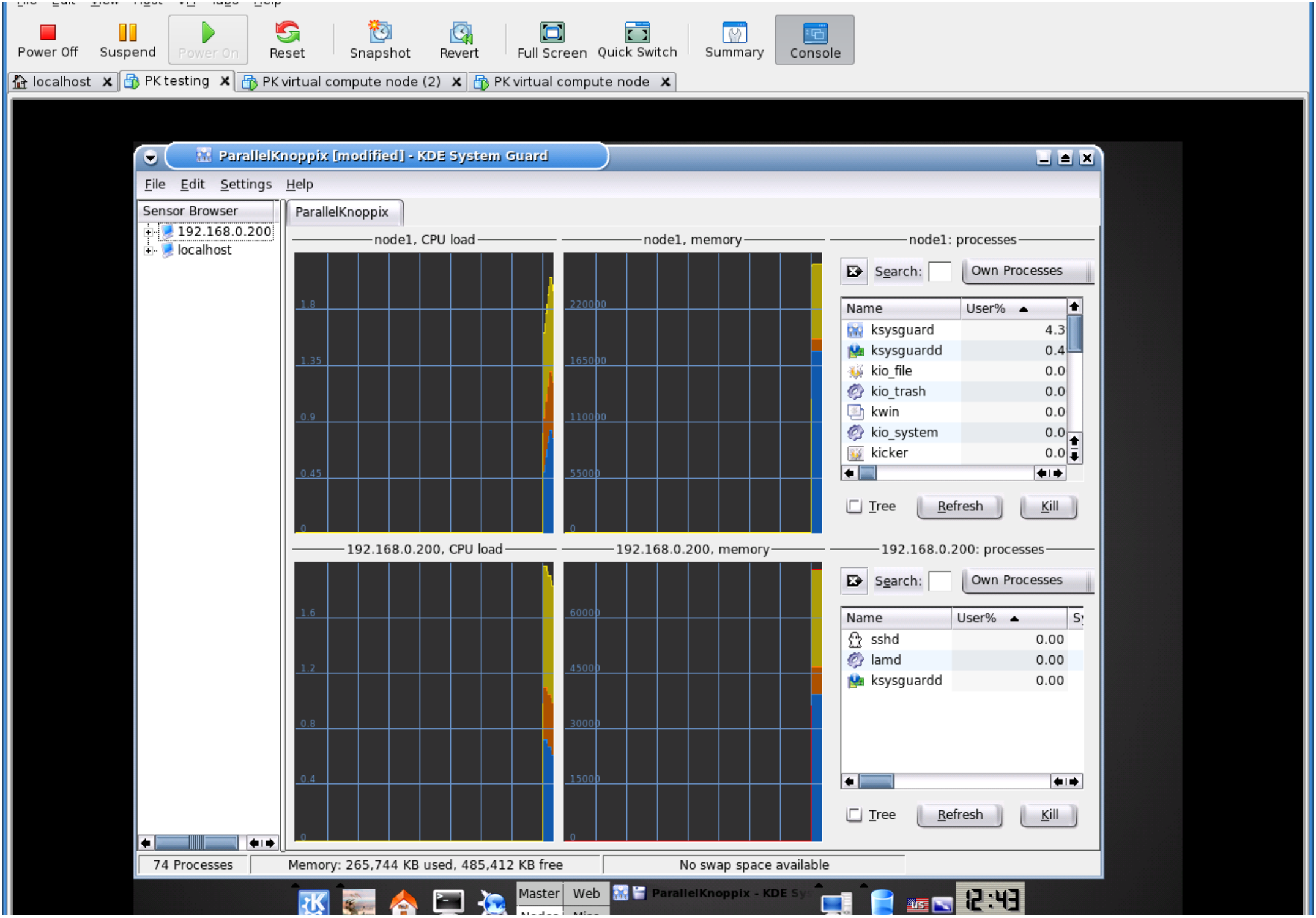
Note that the first screenshot inside KDE was made at 6:02, and this last one at 6:10. Setting up a cluster really is pretty quick, even taking the time to make screenshots as you do it! To see that we really have a cluster running, we can start the monitor:

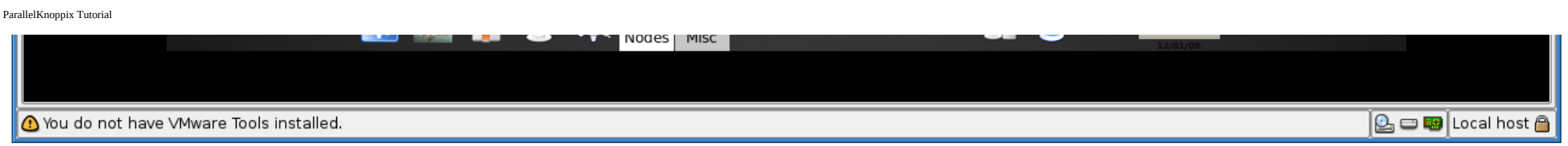




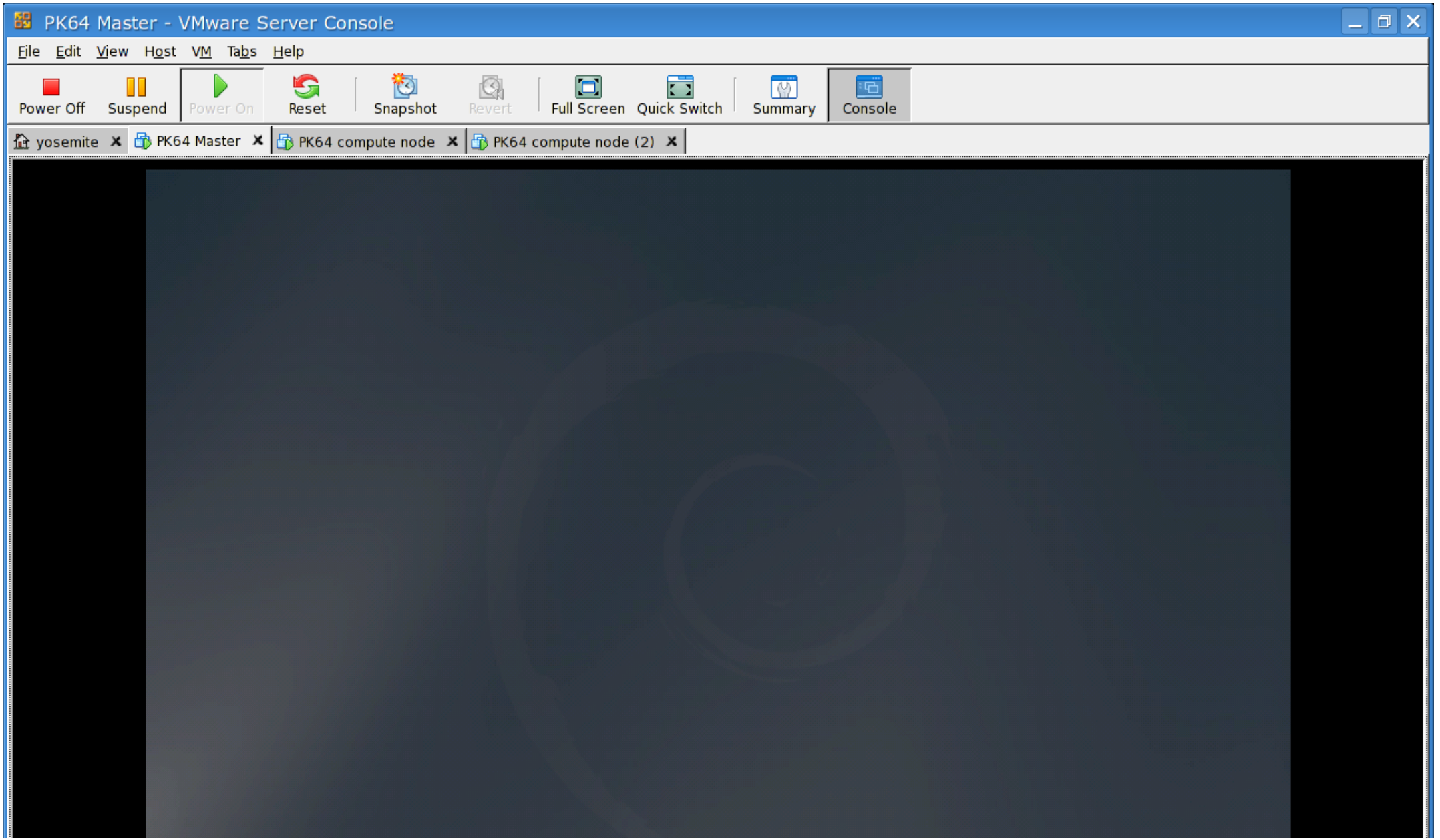
You can see the main node (localhost) and the compute node (192.168.0.200), with a bit of activity:

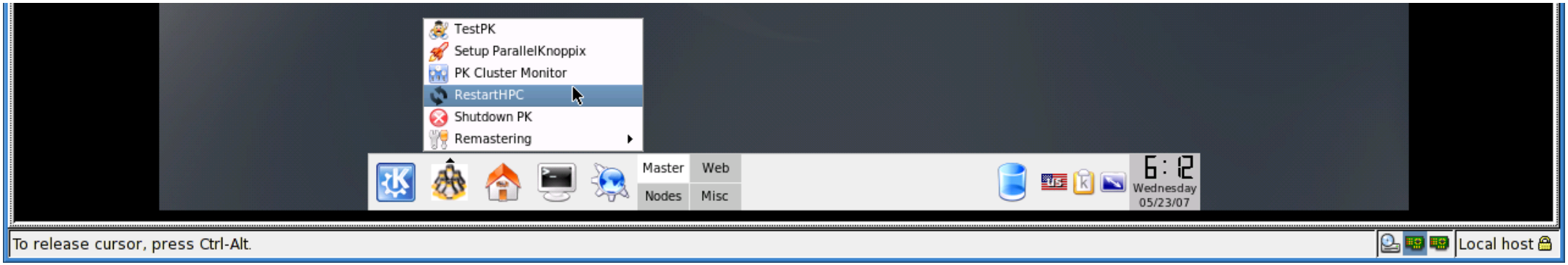



Power Off
Suspend
Power On
Reset
Snapshot
Revert
Full Screen
Quick Switch
Summary
Console
localhost
PK testing
PK virtual compute node (2)
PK virtual compute node
ParallelKnoppix [modified] - KDE System Guard
File Edit Settings Help
Sensor Browser
192.168.0.200
localhost
ParallelKnoppix
node1, CPU load
node1, memory
node1: processes
Search:
Own Processes
Name
User%
ksysguard 4.3
ksysguardd 0.4
kio_file 0.0
kio_trash 0.0
kwin 0.0
kio_system 0.0
kicker 0.0
Tree
Refresh
Kill
192.168.0.200, CPU load
192.168.0.200, memory
192.168.0.200: processes
sshd 0.00
lamd 0.00
ksysguardd 0.00
74 Processes
Memory: 265,744 KB used, 485,412 KB free
No swap space available
Master
Web
12:43

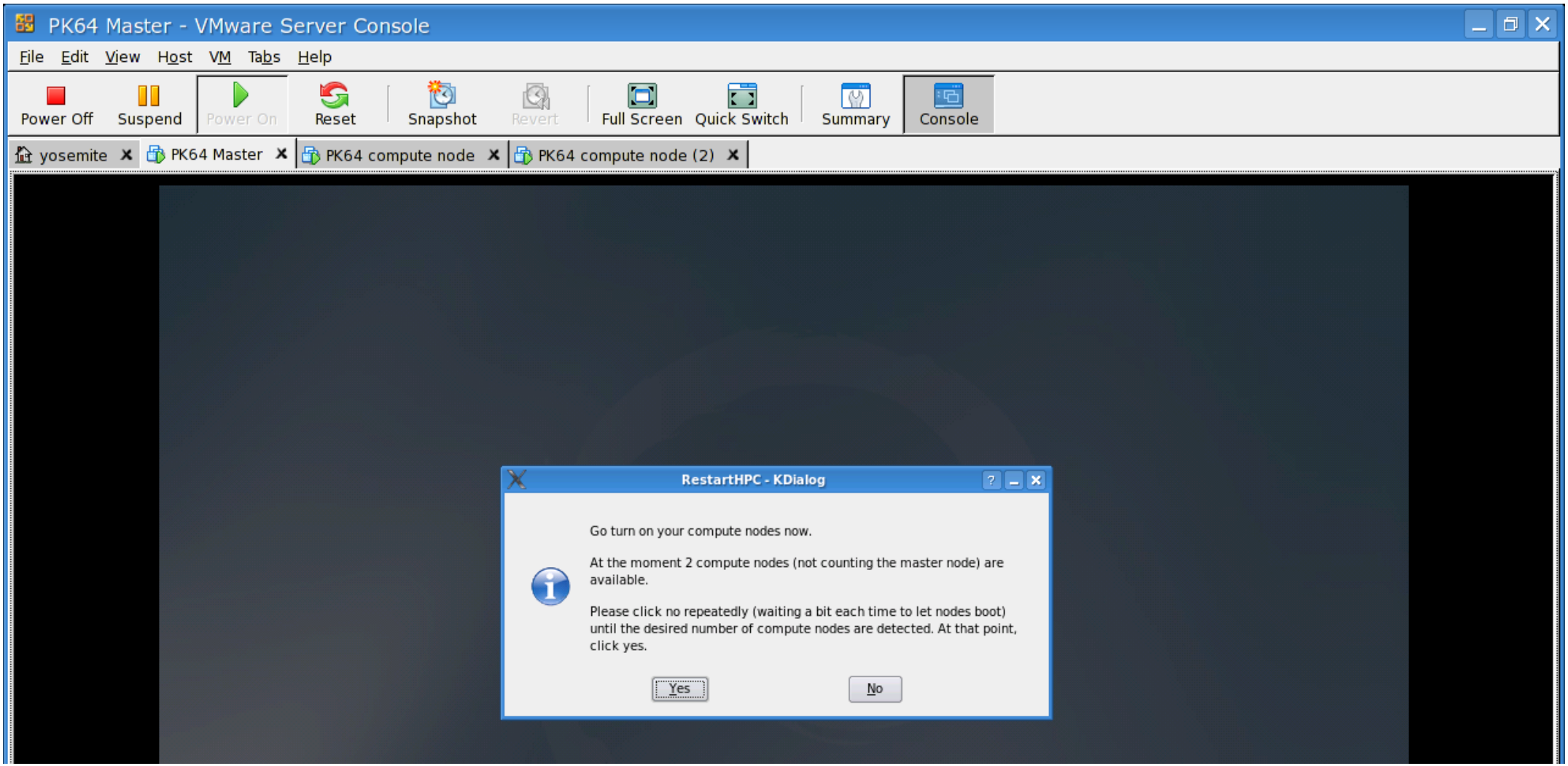


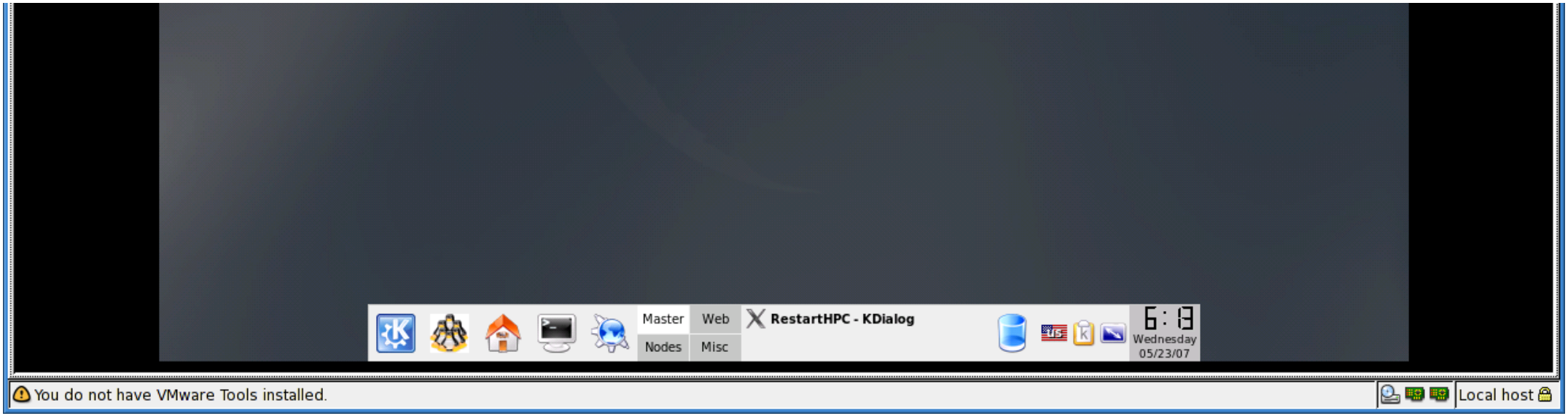
To add nodes, use "RestartHPC":



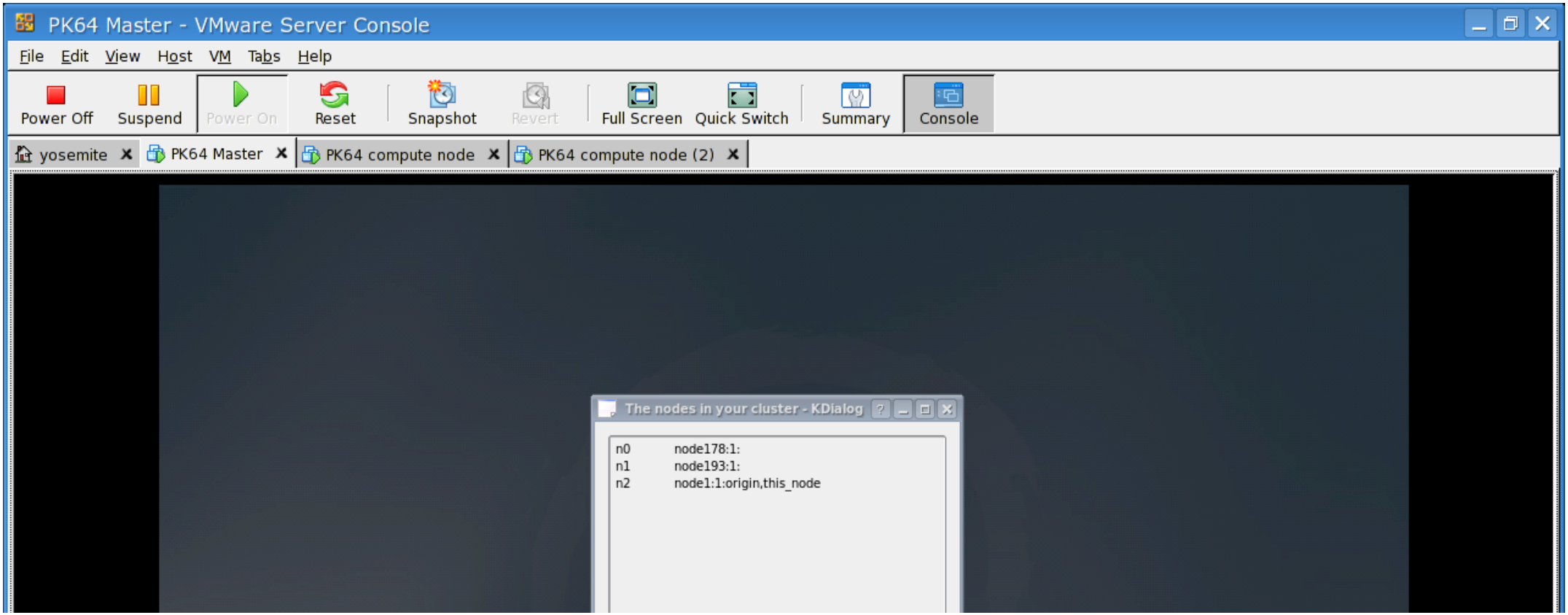


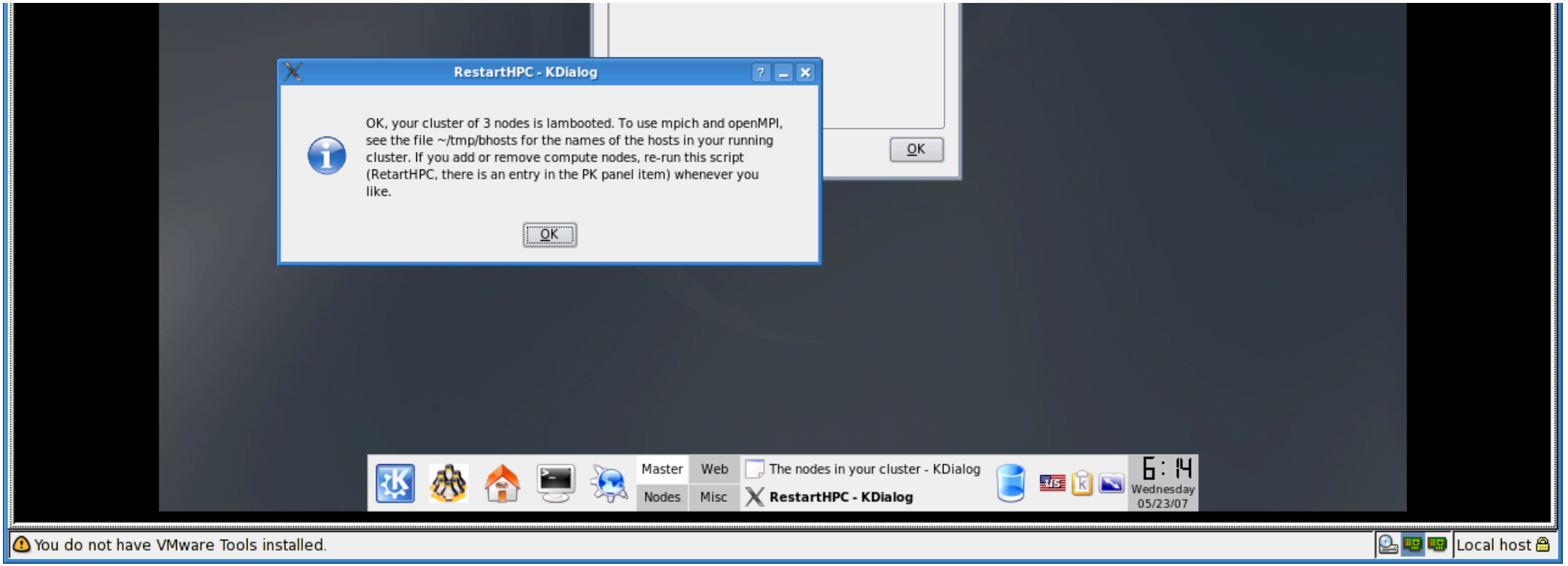
Another node has come up, and we can resize the cluster if we like:





OK, we now have 3 nodes total:



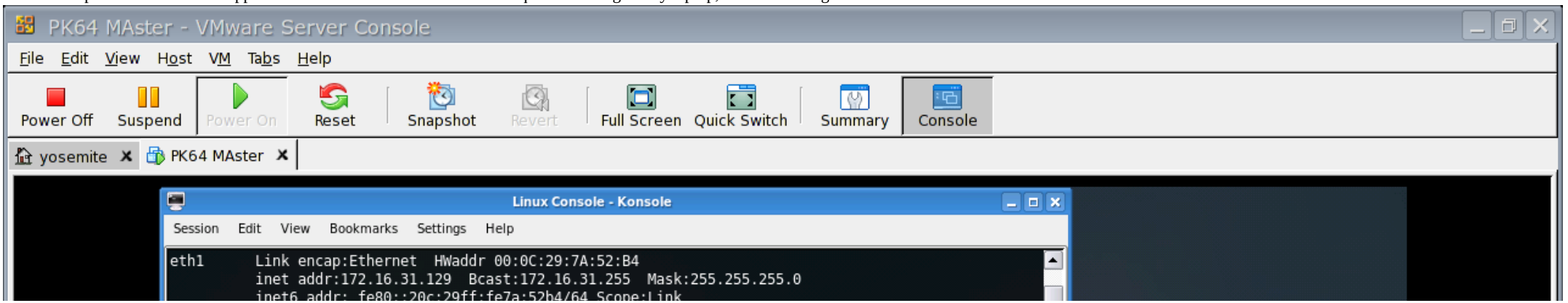


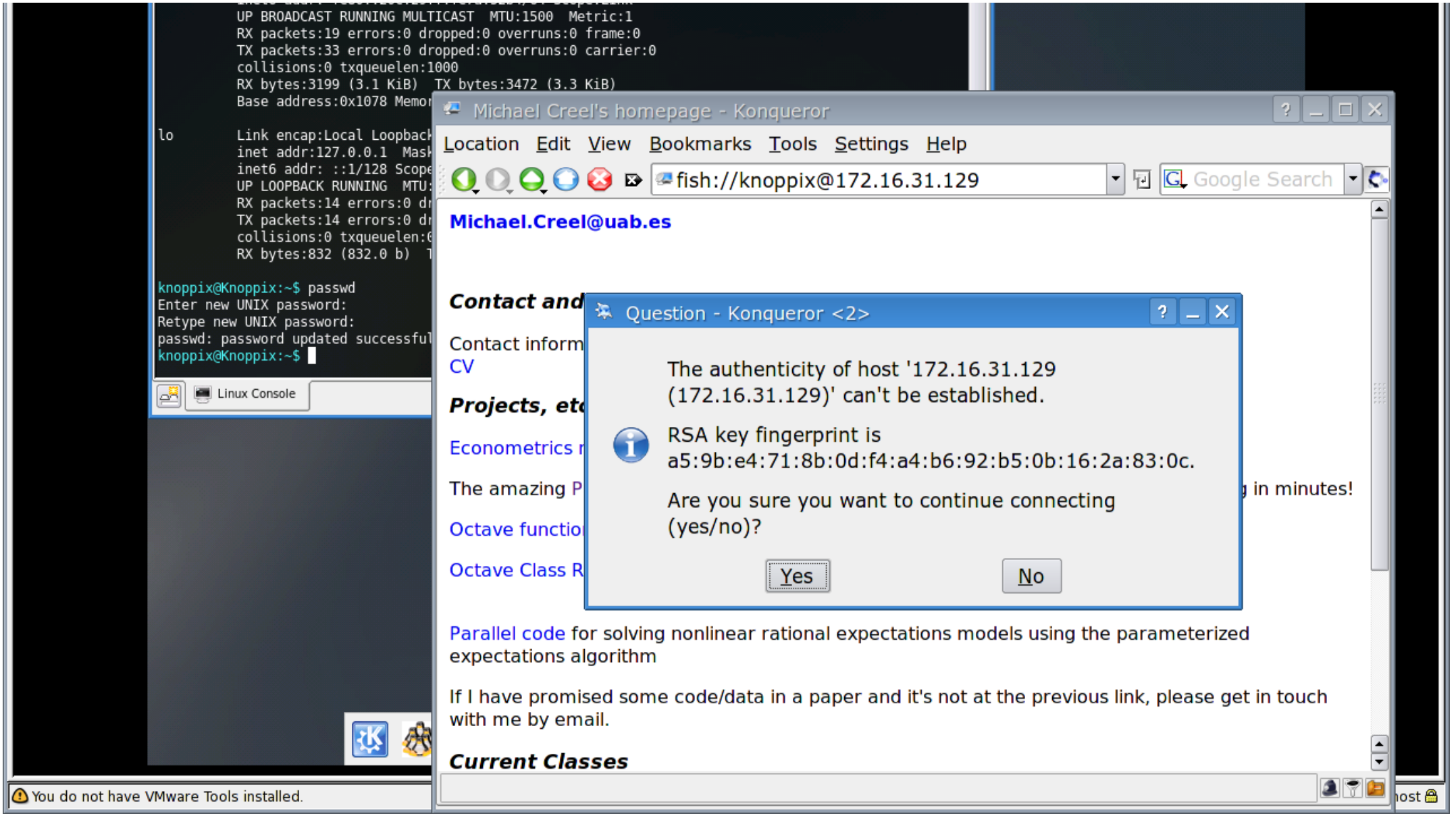
back to index

Use

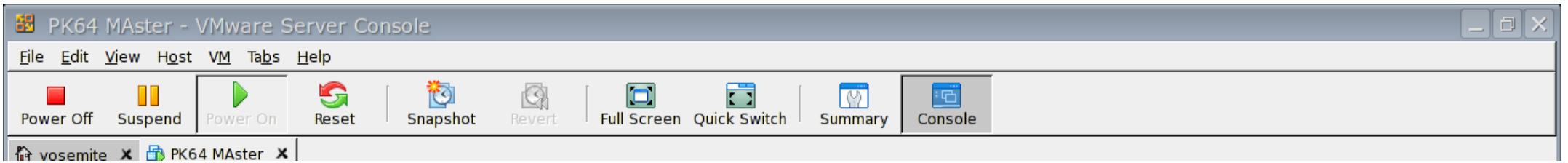
Moving files and data on/off the cluster

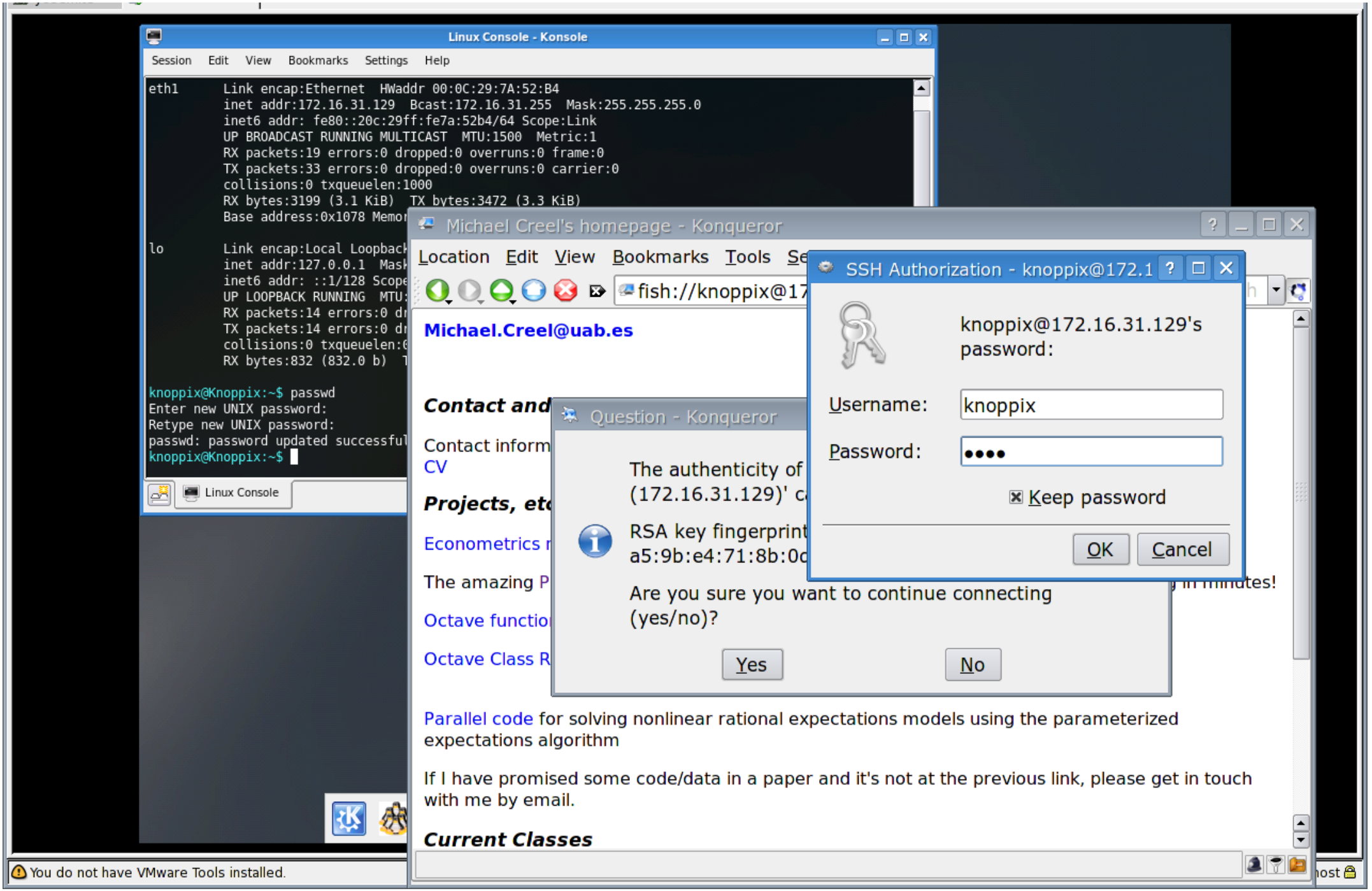
You may be wondering how to get your data and programs onto the cluster. This is pretty easy, assuming that your master node has 2 networking devices (one for the cluster, and one to connect to the rest of the world). In the following shot, there is a VMware Server Console window that shows PK64 running. Note that the IP address 172.16.31.129 is active. This IP address is available to connect the PK cluster to the rest of the world. Also note that I have set a password for the "knoppix" user. The other window shows Konqueror running on my laptop, and I'm fish'ing into the PK cluster.



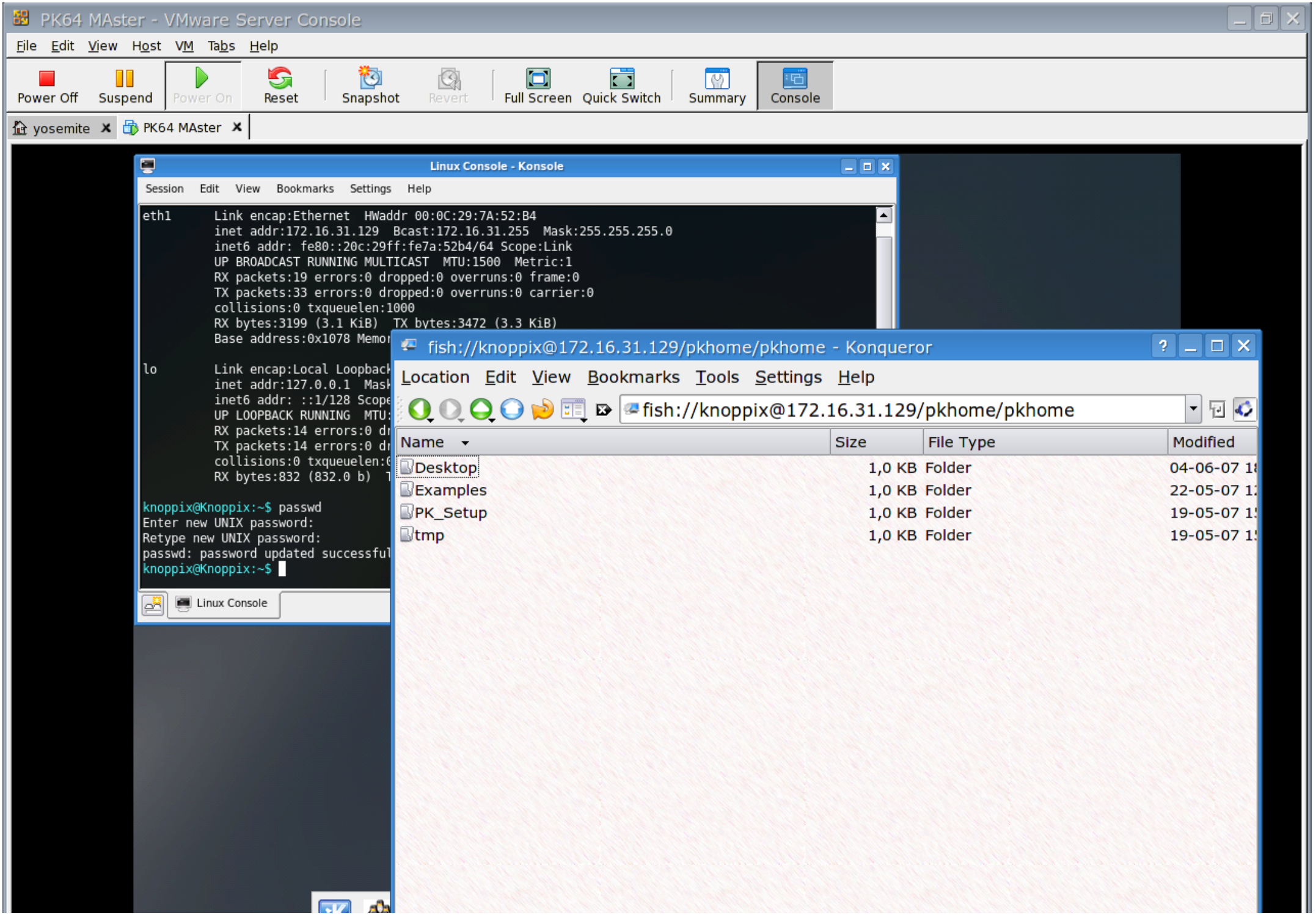


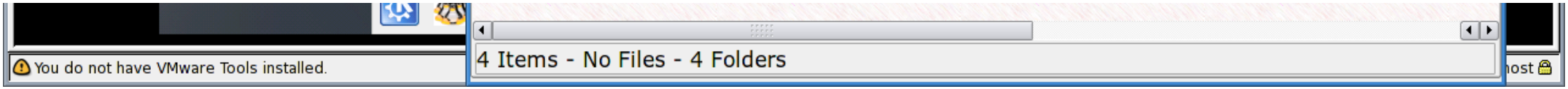
Enter that password:





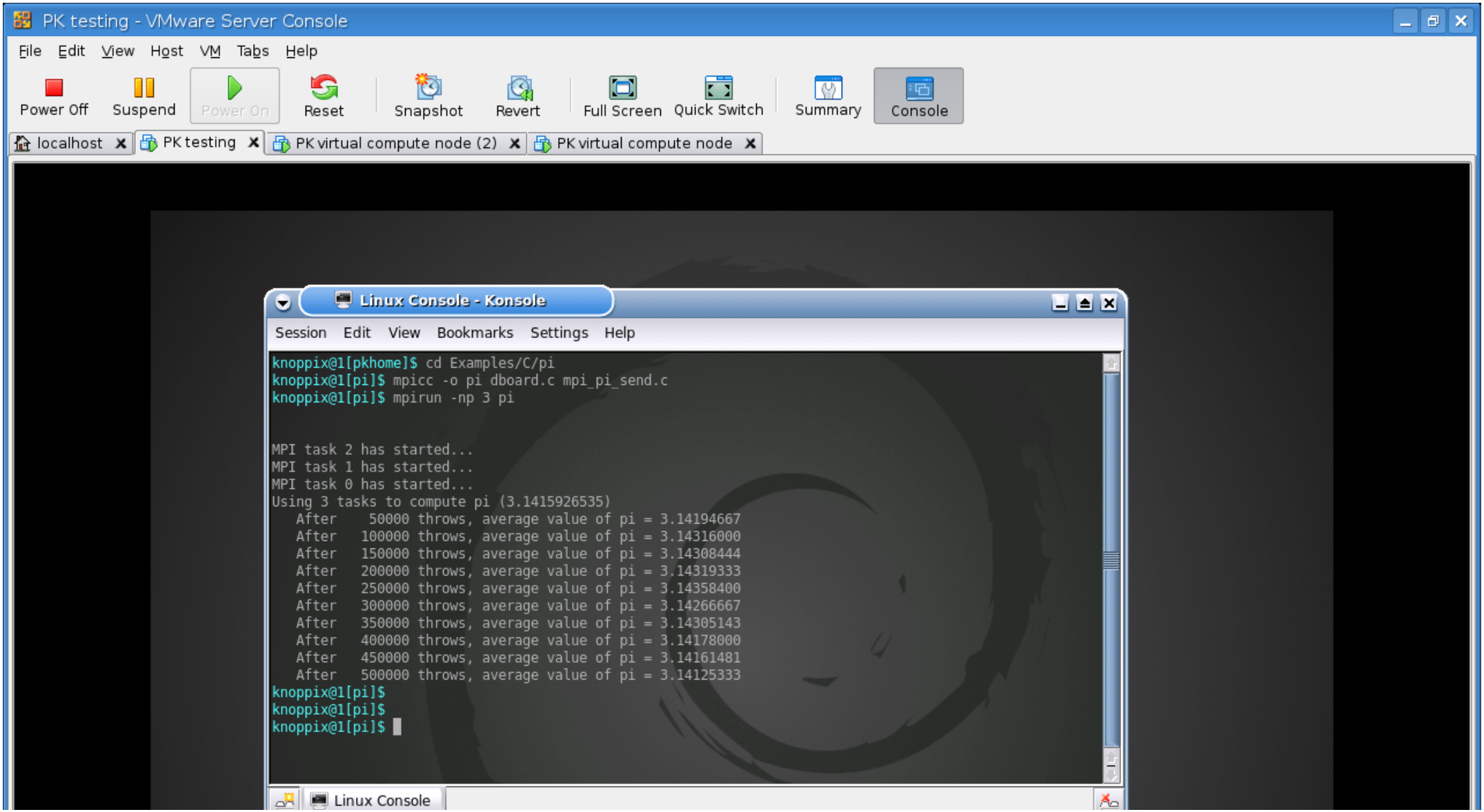
Bingo, I can now do drag'n'drop file administration between my laptop and the PK cluster:


PK64 MAster - VMware Server Console
File Edit View Host VM Tabs Help
Power Off Suspend Power On Reset Snapshot Revert Full Screen Quick Switch Summary Console
yosemite
PK64 MAster
Linux Console - Konsole
Session Edit View Bookmarks Settings Help
eth1 Link encap:Ethernet HWaddr 00:0C:29:7A:52:B4
inet addr:172.16.31.129 Bcast:172.16.31.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe7a:52b4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:19 errors:0 dropped:0 overruns:0 frame:0
TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3199 (3.1 KiB) TX bytes:3472 (3.3 KiB)
Base address:0x1078 Memor
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask
inet6 addr: ::1/128 Scope
UP LOOPBACK RUNNING MTU:
RX packets:14 errors:0 dr
TX packets:14 errors:0 dr
collisions:0 txqueuelen:0
RX bytes:832 (832.0 b) T
knoppix@Knoppix:~$ passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successful
knoppix@Knoppix:~$
Linux Console
fish://knoppix@172.16.31.129/pkhome/pkhome - Konqueror
Location Edit View Bookmarks Tools Settings Help
fish://knoppix@172.16.31.129/pkhome/pkhome
Name Size File Type Modified
Desktop 1,0 KB Folder 04-06-07 18
Examples 1,0 KB Folder 22-05-07 12
PK_Setup 1,0 KB Folder 19-05-07 15
tmp 1,0 KB Folder 19-05-07 15



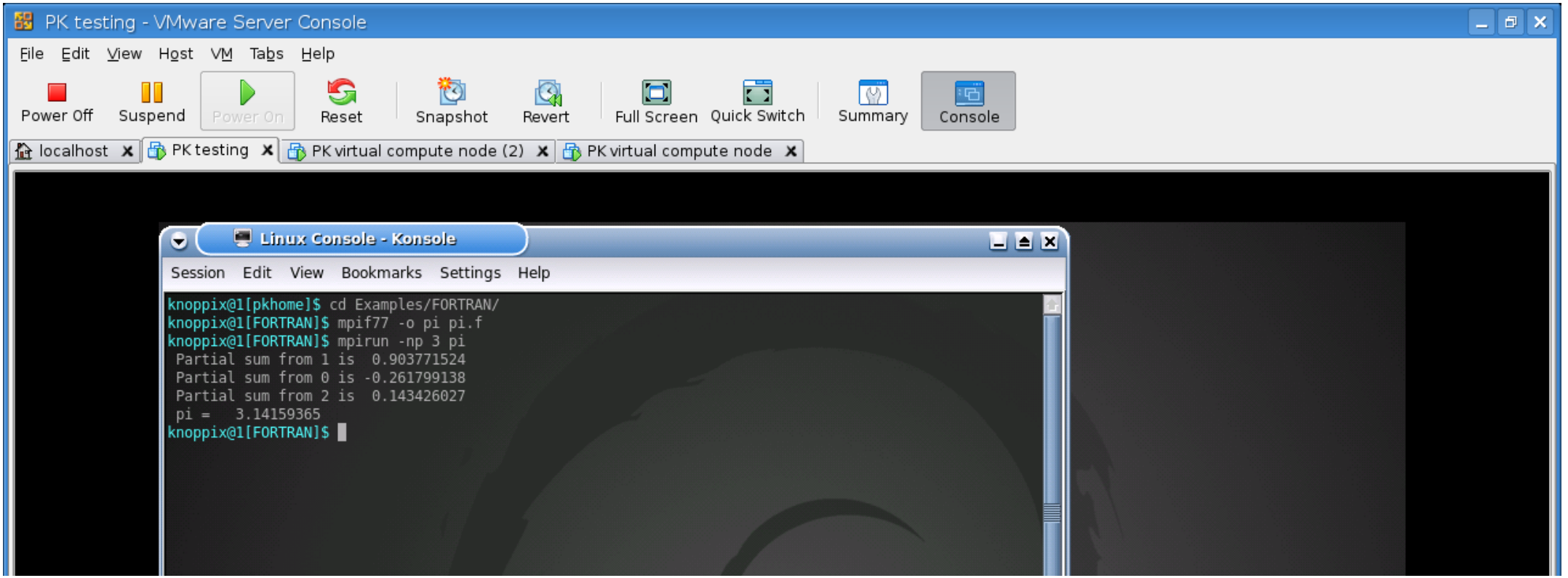
If your master node has only 1 networking device, you can use hard disk partitions, USB storage devices, etc., to move things around. Or you can decide to use a virtual machine for the master node. Or you can go buy another net card.
Back to index

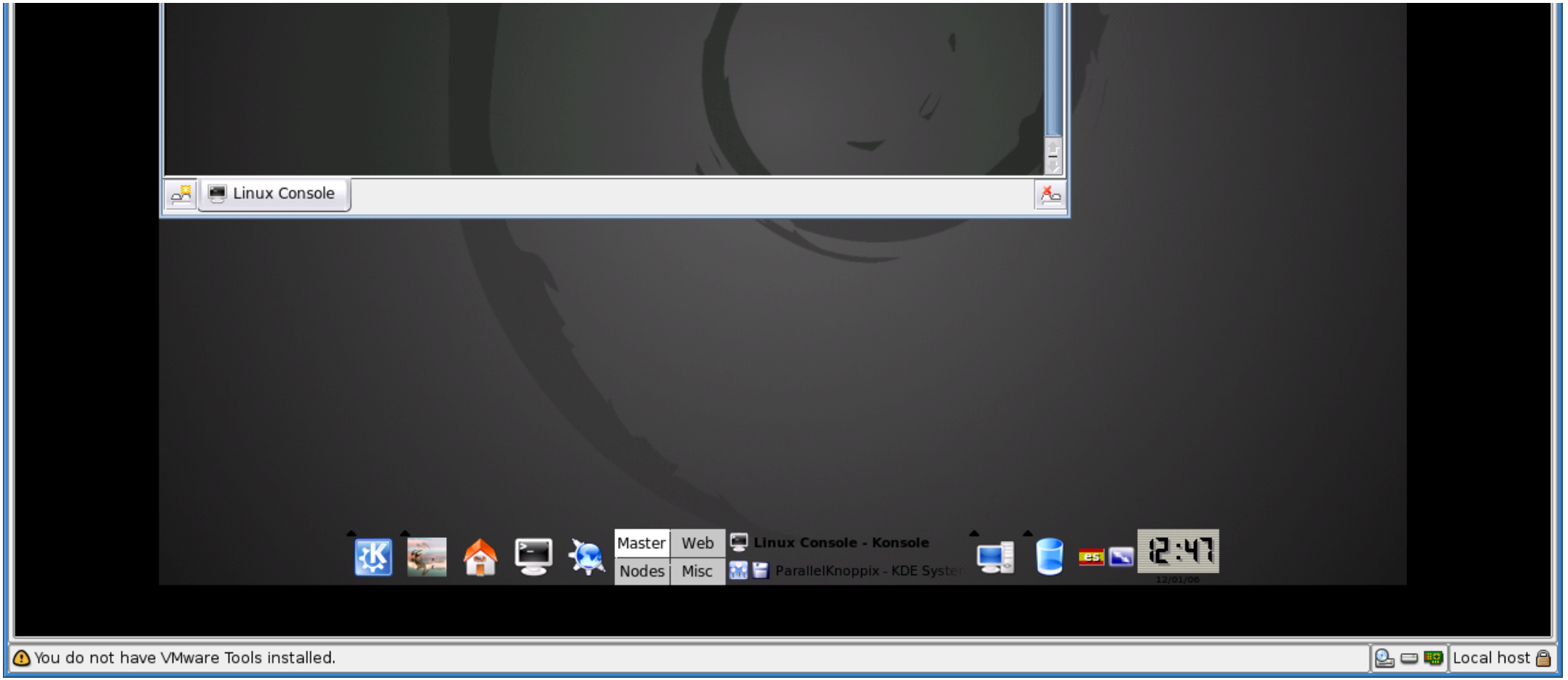
An example of compiling and running an MPI program in C:



Back to index

An example of compiling and running an MPI program in Fortran:

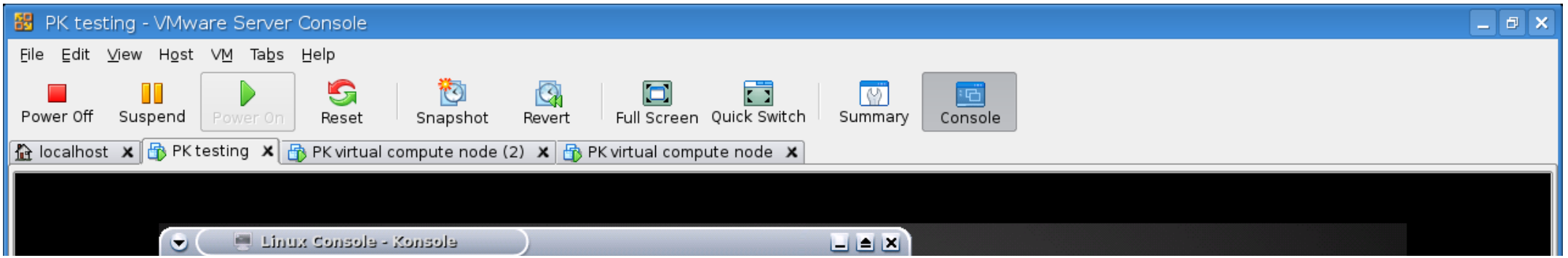


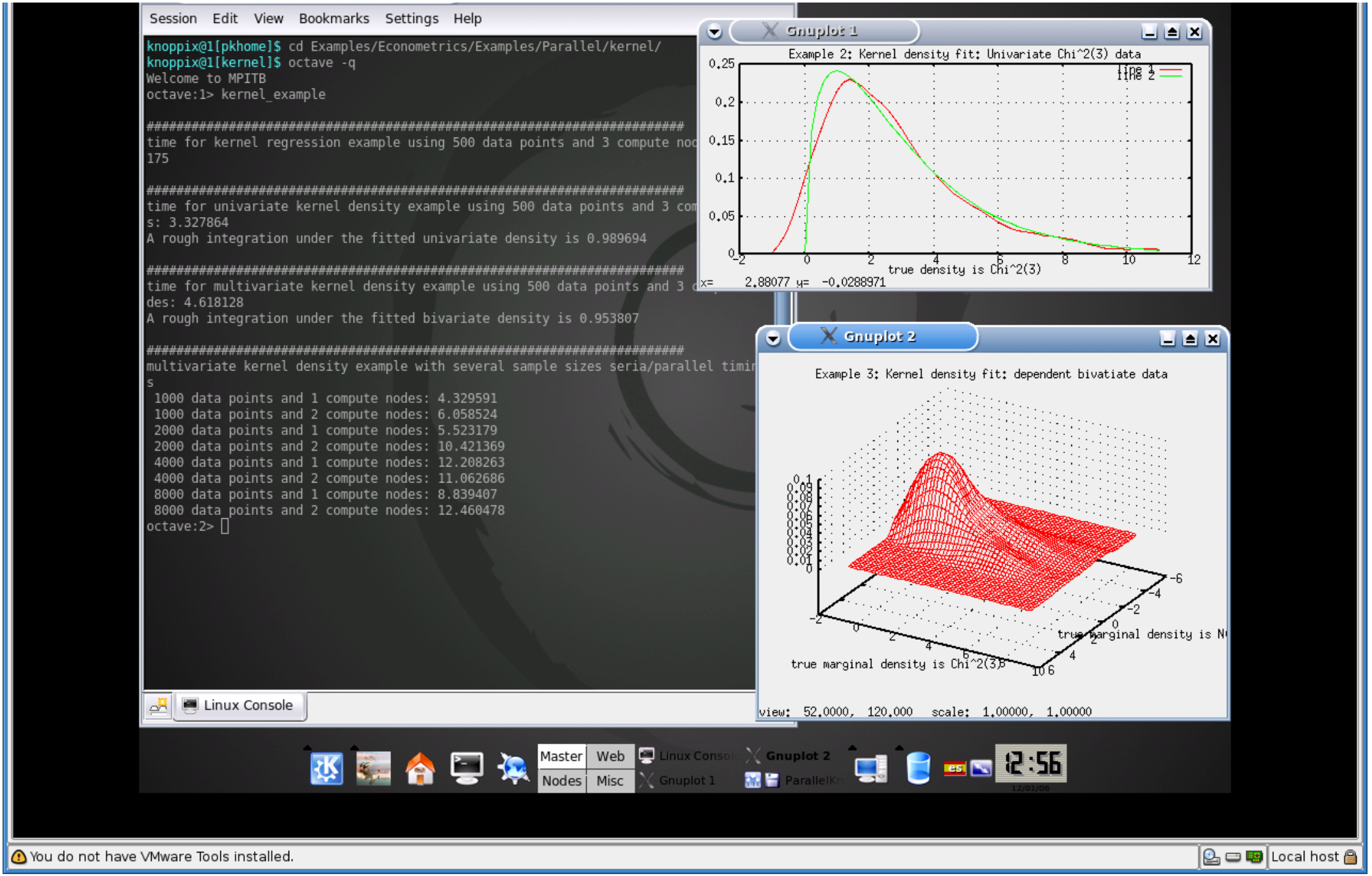


Back to index

An example using MPITB for GNU Octave:

Please note that the timings here don't show a speedup when nodes are added since this was done on a virtual cluster (3 virtual nodes) that runs on a single real machine.

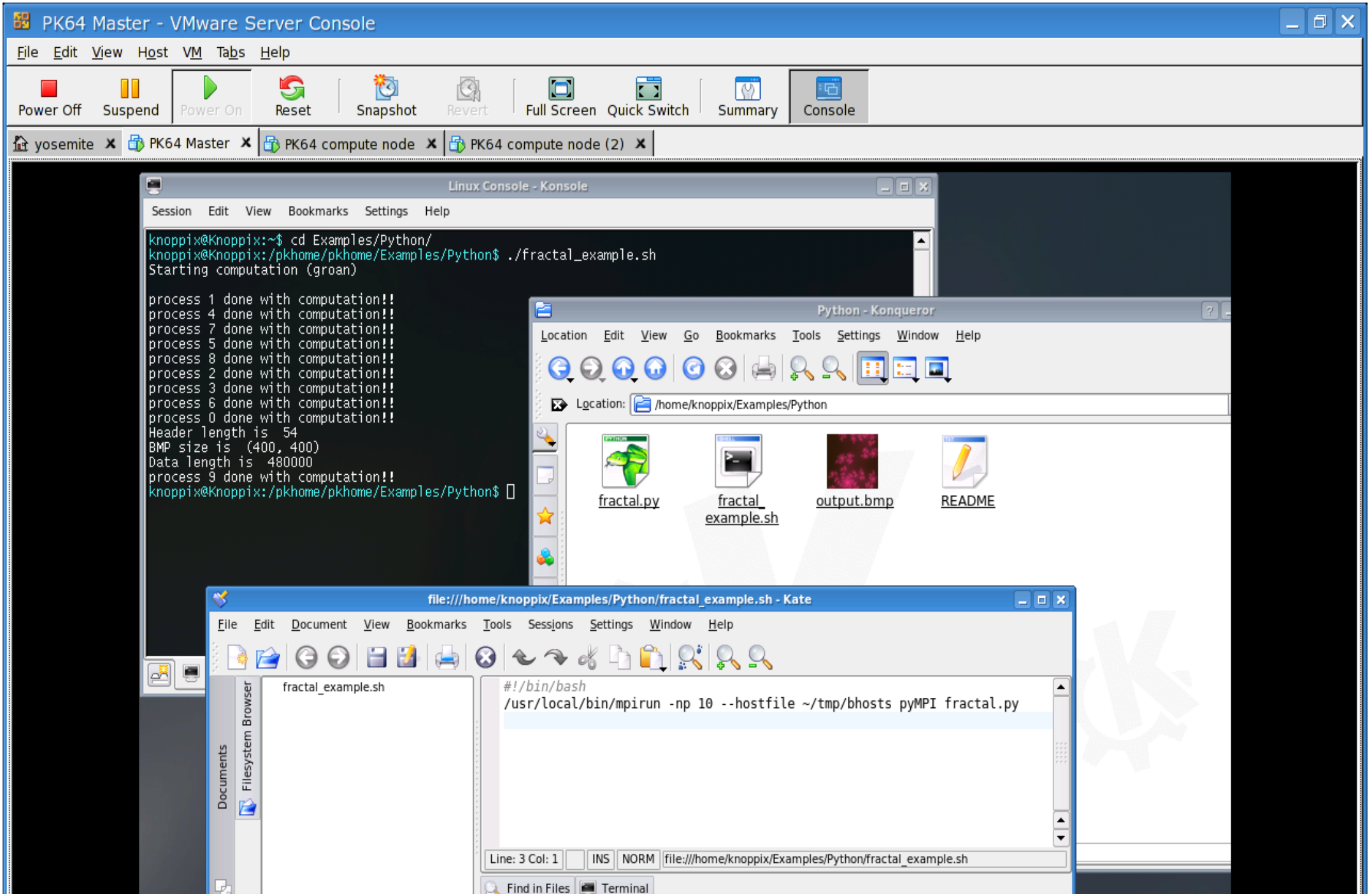


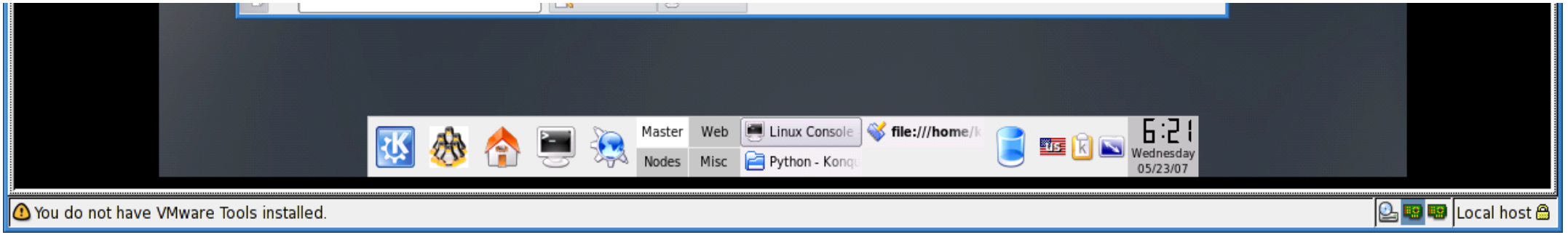


There are other examples using MPITB for Octave in the subdirectories inside /home/knoppix/Examples/Econometrics/Examples/Parallel

An example using Python (pyMPI):

What's a clustering distro without a fractal plotting example - NOTHING! So here we go:





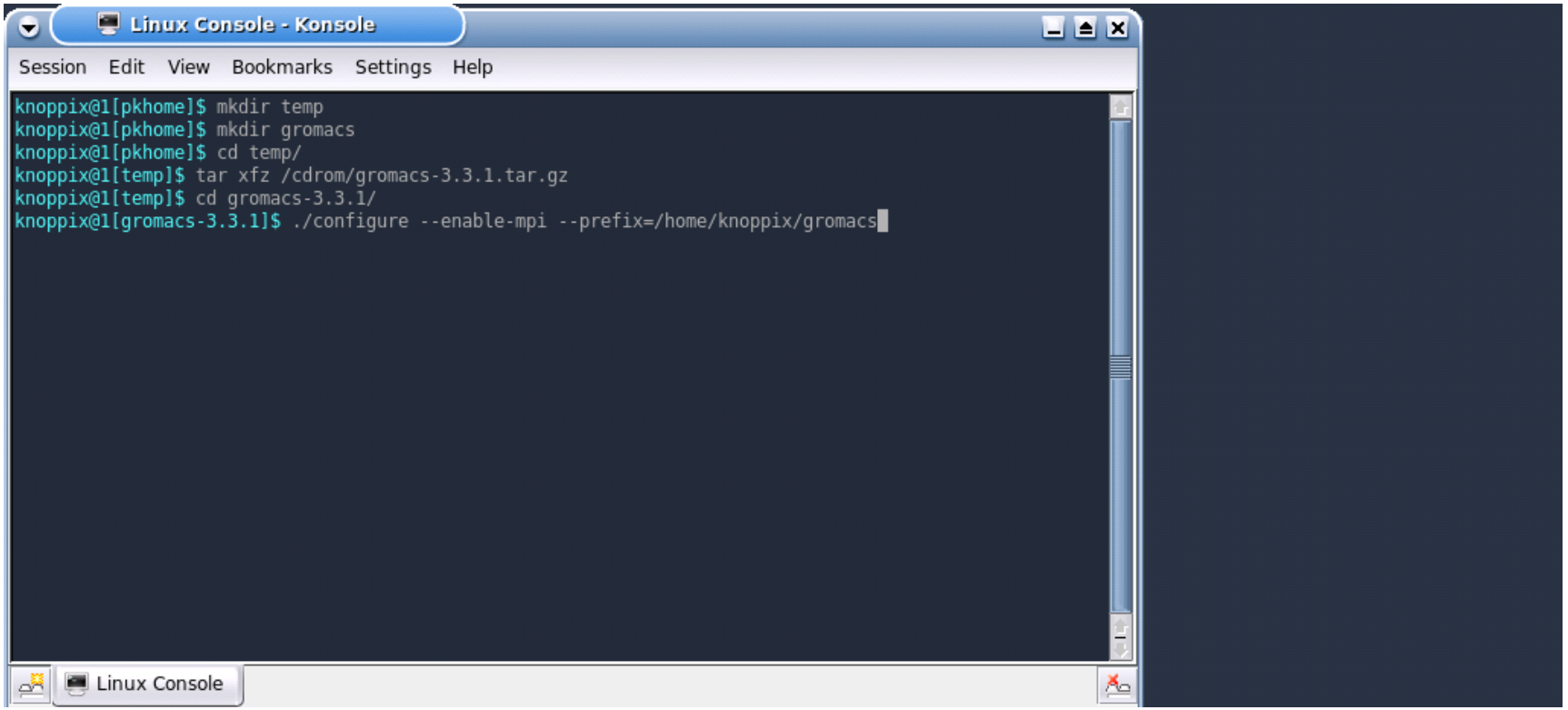
Back to index

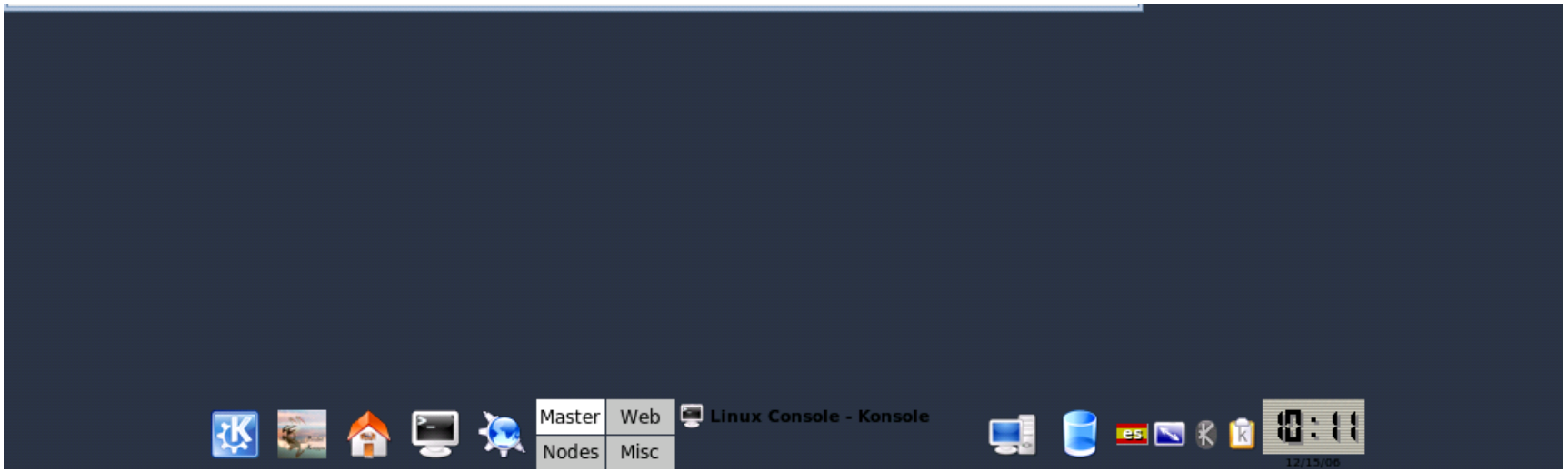
How to add software without remastering

You probably will want to use programs that are not on the PK CD. You can modify the CD to add packages (a process known as re-mastering the CD image) but in most cases it's easiest to just install the software inside the home directory of the knoppix user. If you do this, you might want to make a backup copy of `/pkhome/pkhome` (the real location of `/home/knoppix`, which is a symbolic link) before shutting down, since this is on a ramdisk that goes "poof!" when the master node is turned off. See the section on file transfer for how to copy your backups off of the cluster.

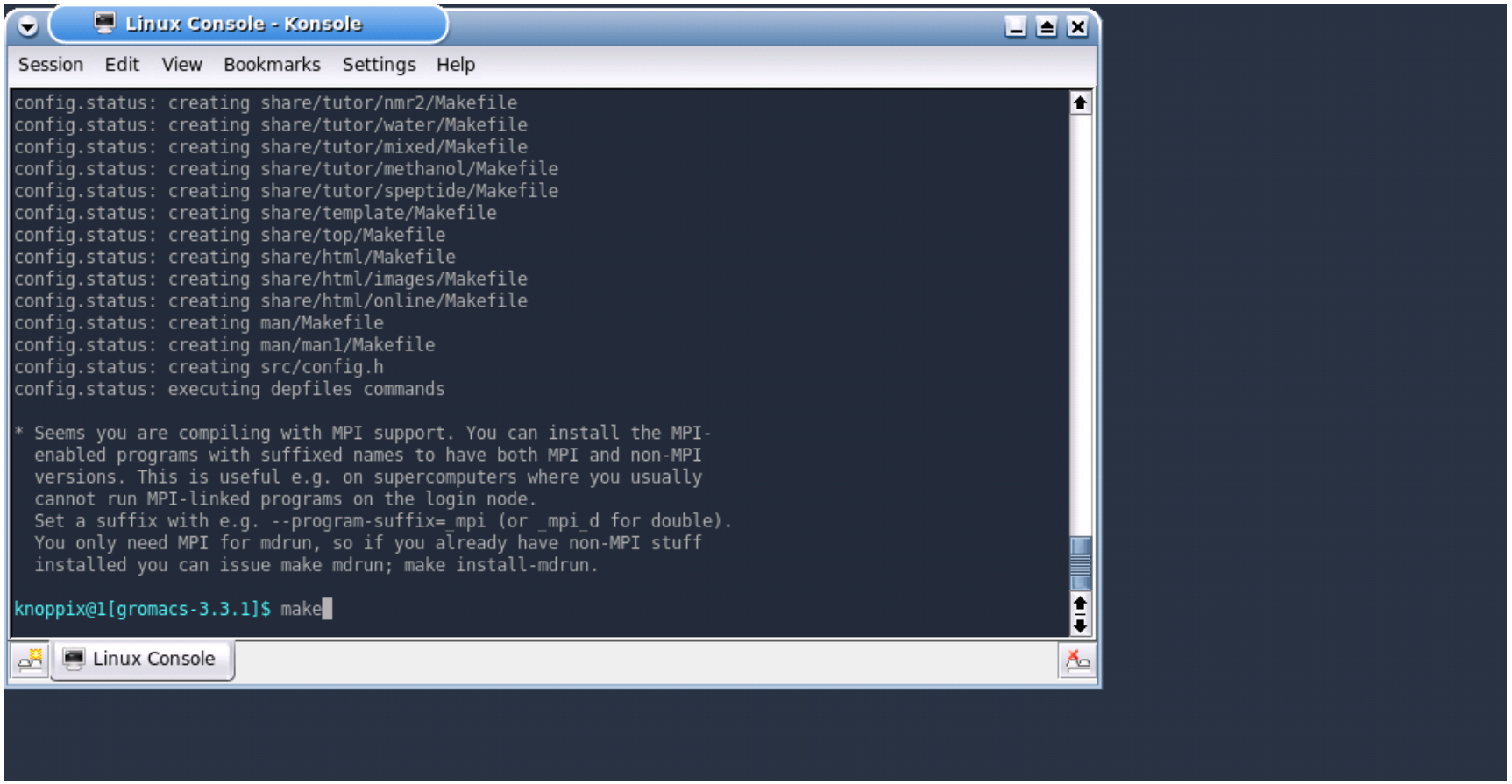
We'll use Gromacs, a program for molecular dynamics, as an example. **NOTE**: Despite what you see in the next image, you now need to download the source tarball, it's no longer on the CD, for space reasons.

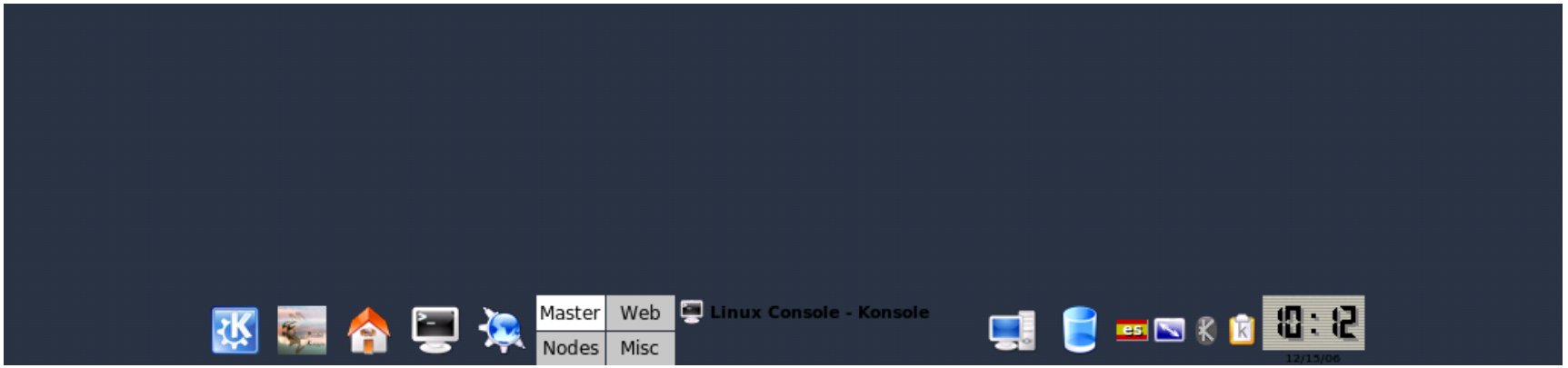
The first step is to make a temporary directory, an installation directory, and to uncompress the source tarball into the temporary dir, and to configure the compilation to use MPI:



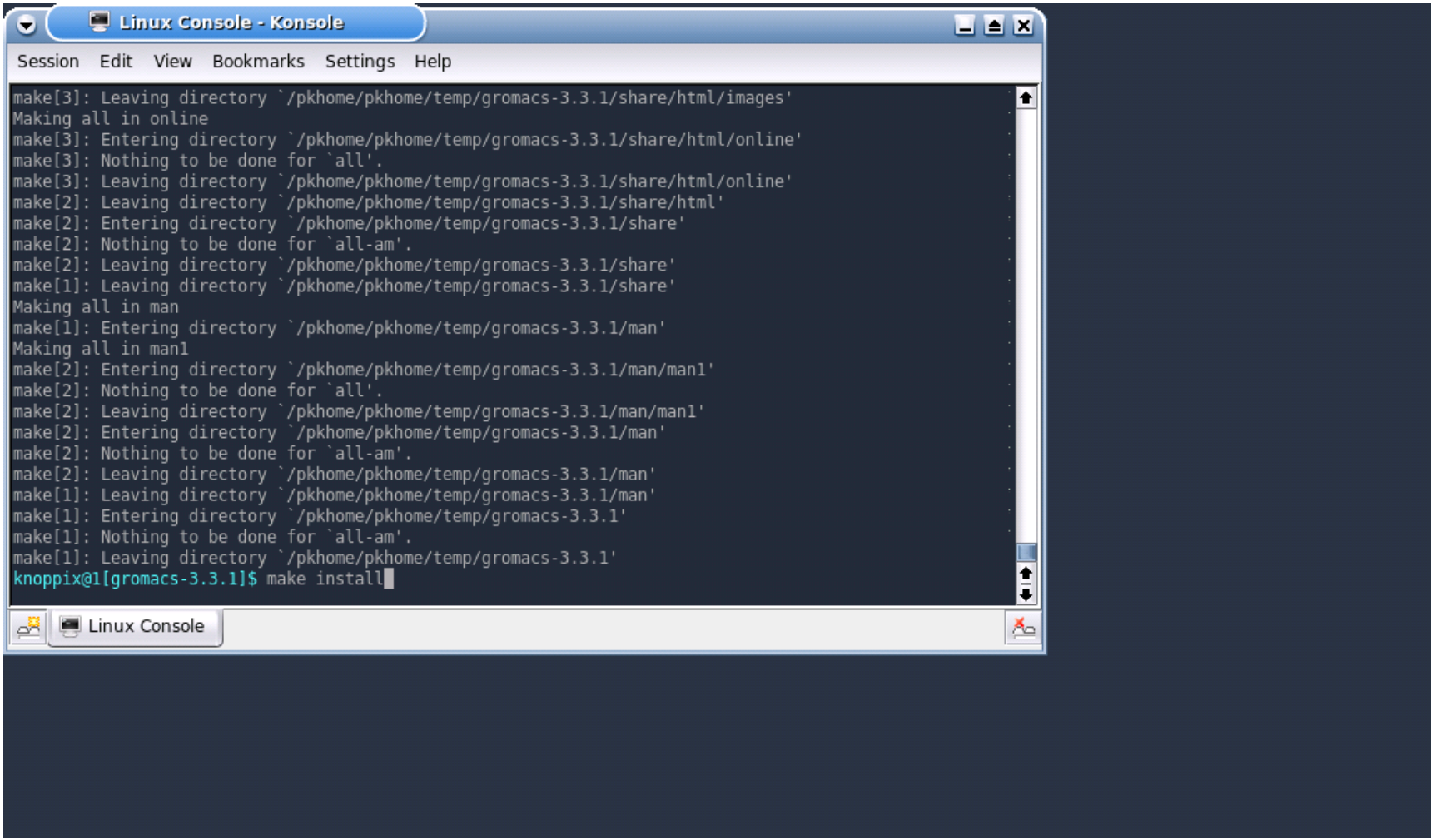


Once you hit enter in that last window, a fair amount of output whizzes by, and you end up seeing the next picture, ready to make the package. Do that:

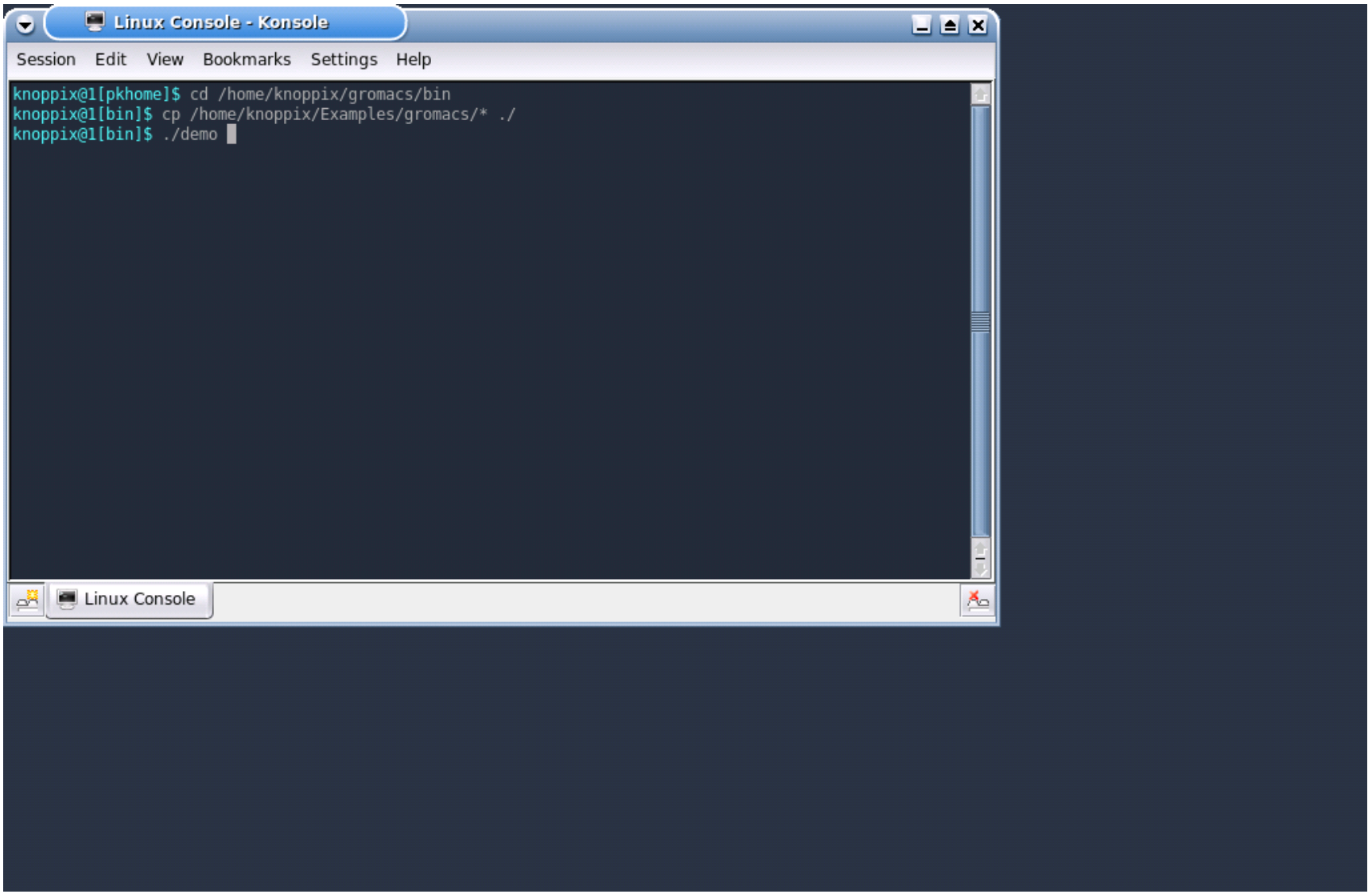




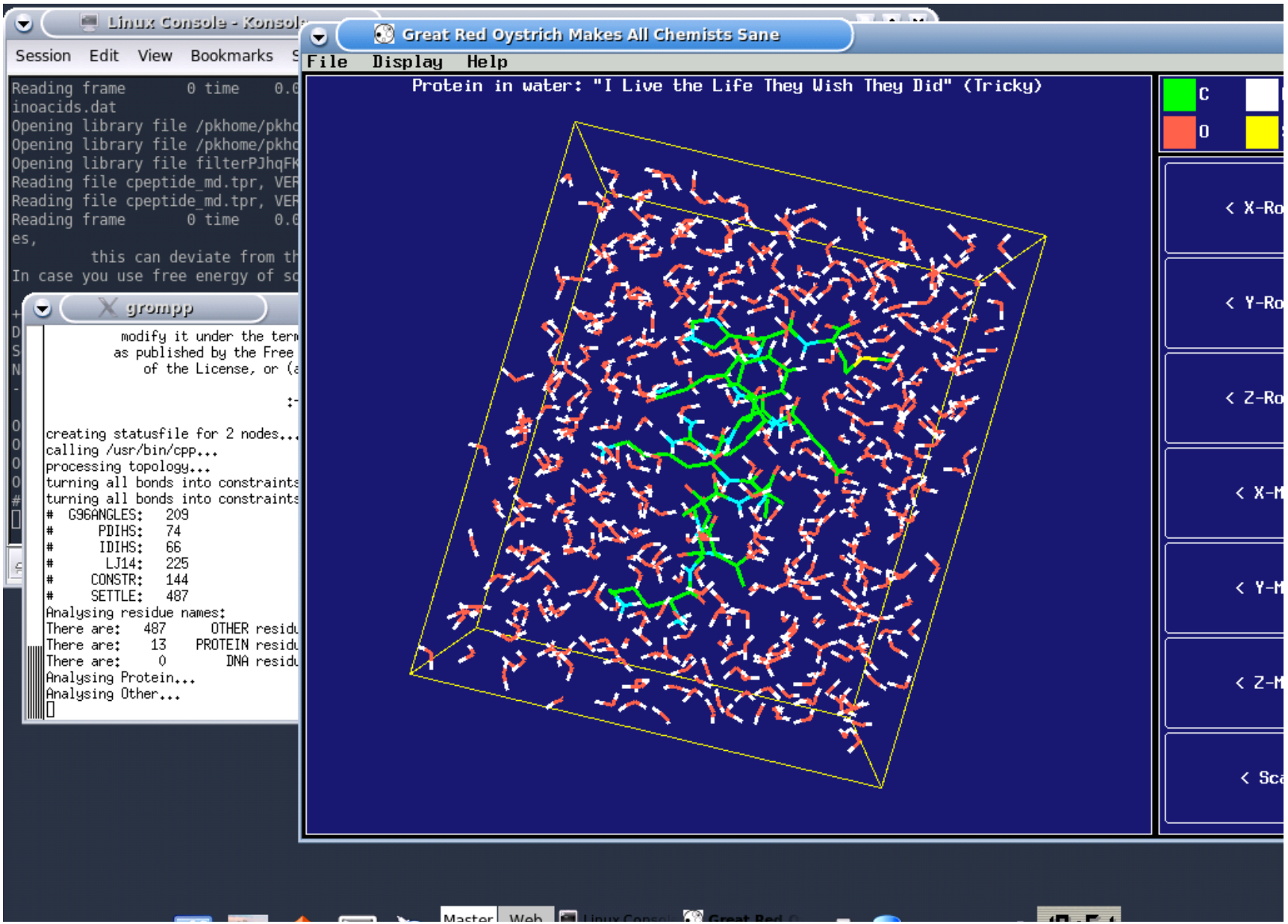
Making Gromacs takes a while. Eventually you arrive to the following screen, ready to make install. Do that:



Next, change to the ./bin subdir of your installation directory, and copy over the demo program (ugh, what a cheesy hack). Then run the demo.



You need to hit enter a number of times, but finally, presto:



OK, that's more or less how to install a package. If you want to use the package the next time you set up the cluster, remember to save the installation directory somewhere, or it will disappear when you shut down. If you want to install software permanently, you will need to remaster the CD.

Back to index

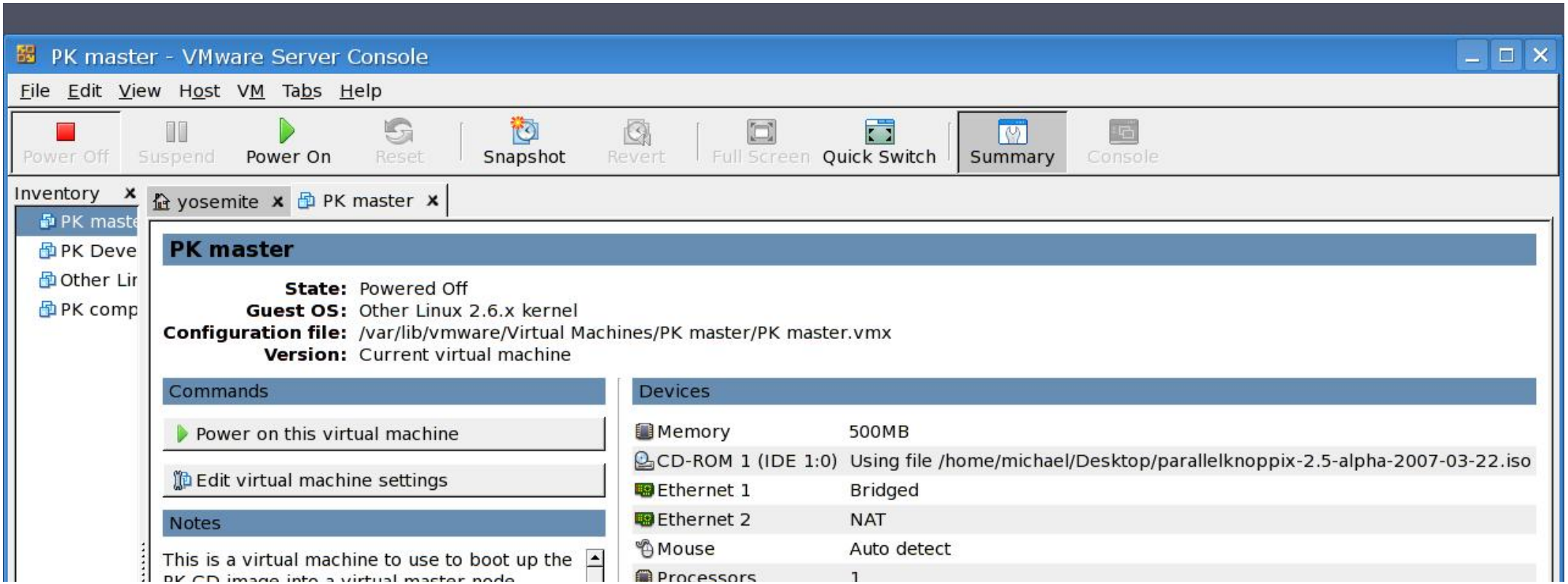
Running PK on another OS using virtualization

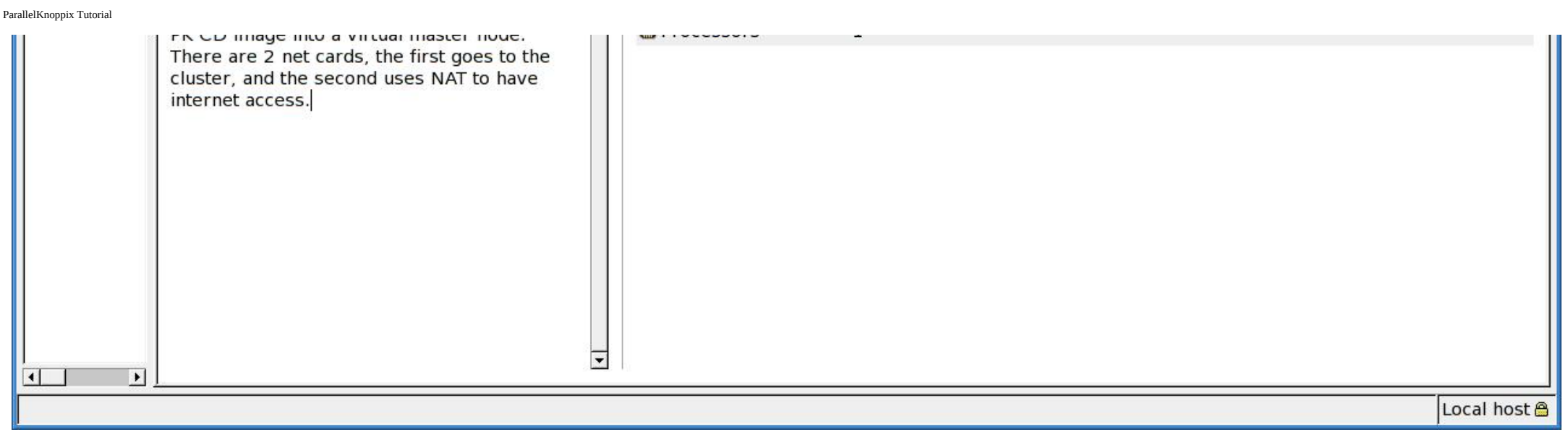
PK can be run on your regular Linux distro, or on other operating systems, inside a virtualization environment (QEMU, VMWare, kvm, etc.). This is most convenient way to use it, in my opinion. You can just download the PK ISO image to a laptop computer, and by following the steps outlined here, you can be ready to create a cluster at any moment, without burning the image to a CD, without rebooting, all from your favorite installed OS.

VMware

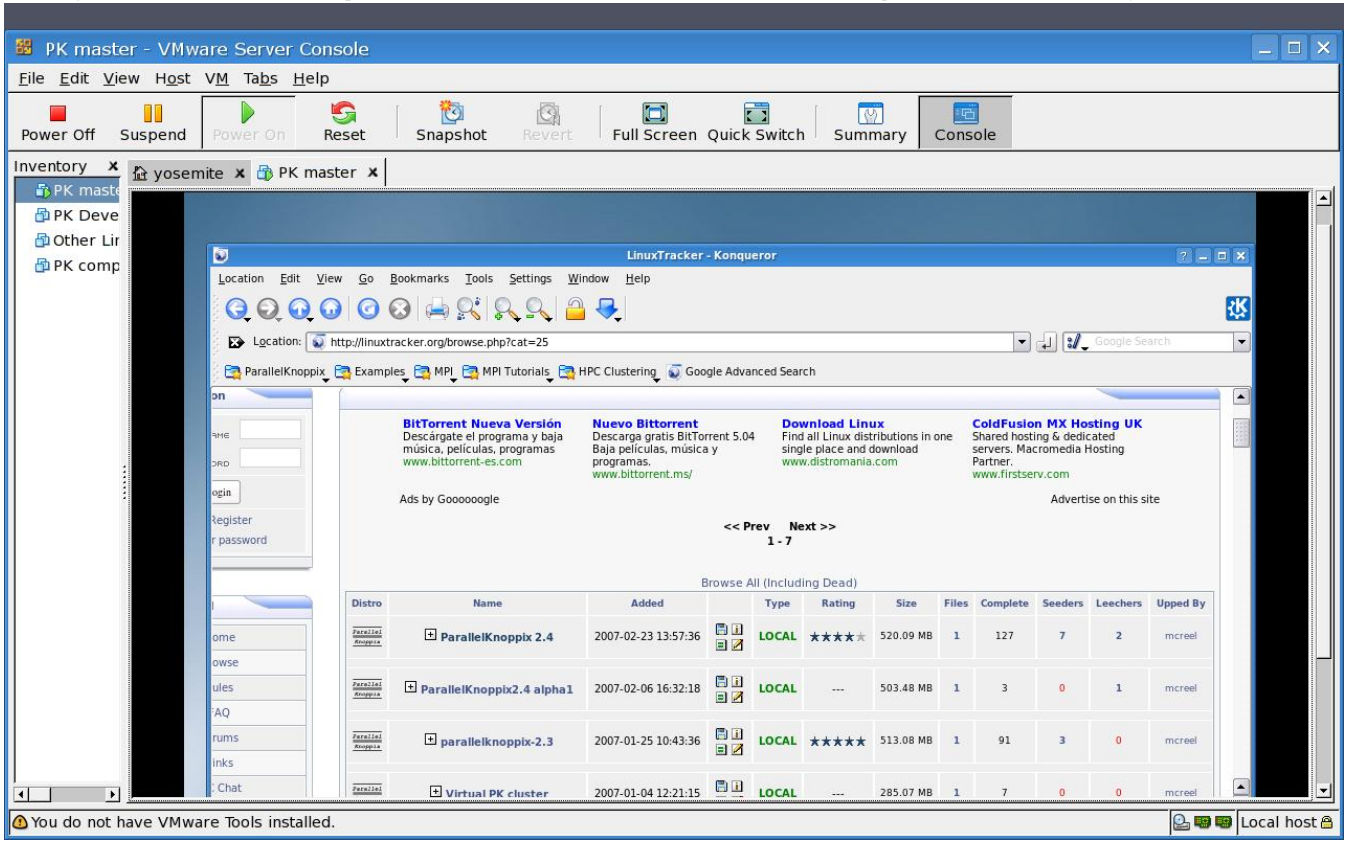
This shows how to use VMware server, v1.0.2 with PK

- First, you need to install VMware Server on your computer. Get it here if you don't have it. This is a free download. Configure VMware to allow bridged and NAT networking when you do the install.
- You also need the latest PK ISO image, saved to some place you can get to while running Windows.
- Next, you need to create a virtual machine. Set it up as in the following shot. Notice that the CDROM device points to the location of the ParallelKnoppix ISO image. You do **not** need to burn the image to a CD. There are 2 network cards, the first uses "bridged", and is for connecting to the cluster. The second uses "NAT", so that the virtual master node will have Internet access. Virtual disk space is not required, but you may prefer to add some depending on what you do with the cluster. Disk space gives you a place to save your work between sessions, for example.



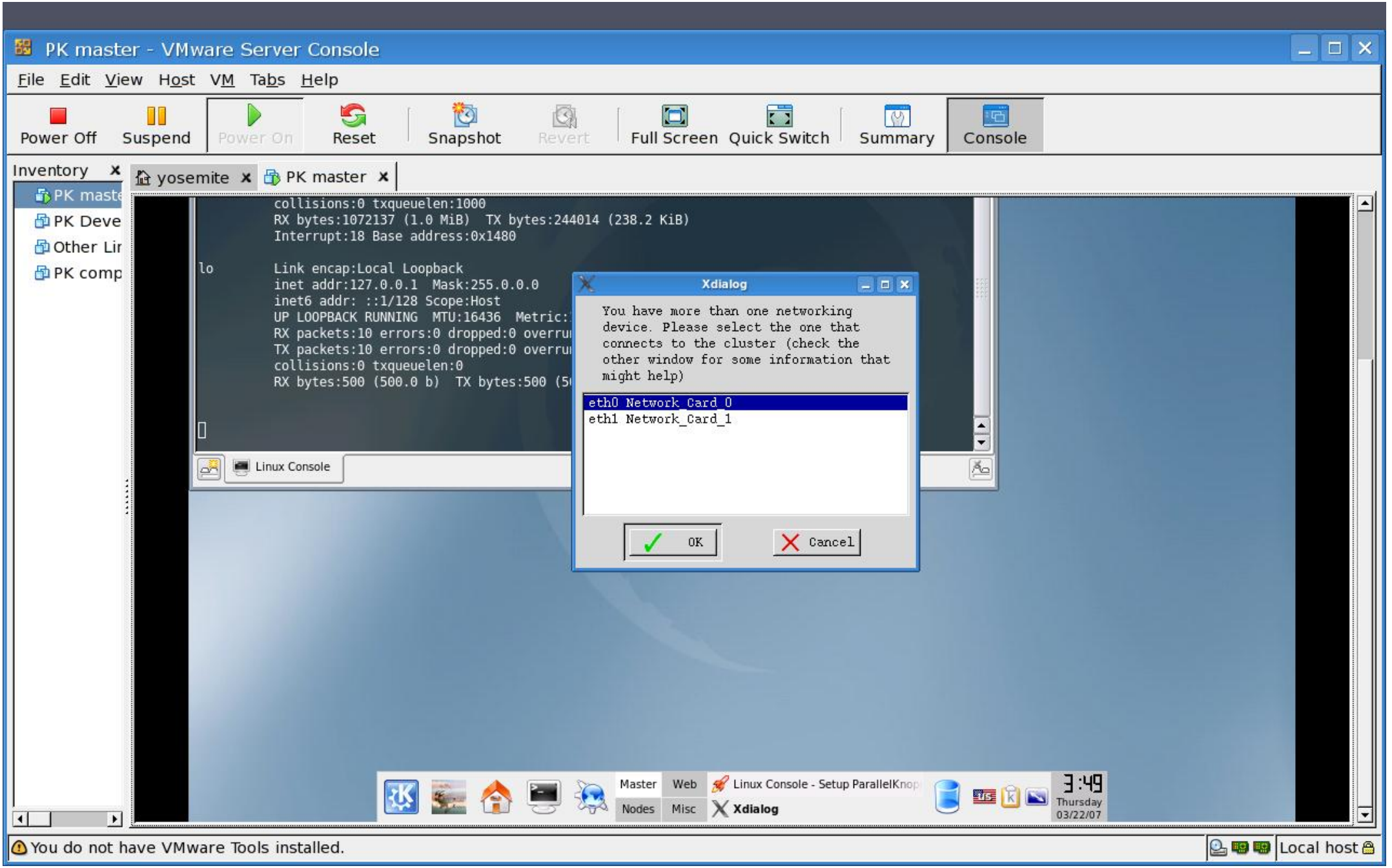


Next, power on the virtual machine. Everything will look just like it does in the initial part of this Tutorial. Note that the bookmarks in the Konqueror brower work, since your second virtual net card is sharing your net connection with your host machine. For example,



OK, that's working :-).

Now, you set up the cluster, just as is explained in the first part of this tutorial. One detail, since you have 2 virtual net cards, you will be prompted to select the one that connects to the cluster, as in the following shot:



You should choose the first one, if you created the virtual machine the way I suggested above. From here on out, just follow the first part of this Tutorial, to finish setting up. Remember that you can make a snapshot when you're done. Then you'll have the PK master node available in seconds, whenever you need it.

An important point: The virtualized master node (192.168.0.1) will probably be slower than your real compute nodes that you PXE booted, since the virtualization overhead is non-negligable. If this is the case, you don't want the master node to enter into your parallel calculations, or the other machines will constantly be waiting for it (unless you write some load balancing into your programs). You need to:

- open a terminal
- ssh into node200
- edit the file /tmp/bhosts, and remove the last line. Then save the file.
- type "lamhalt;lamboot /tmp/bhosts".
- type "exit" to leave ssh on node200 and go back to the master node. That's it! Now all calculations will be on the real nodes, and your virtual master node is just for management and frontend work.

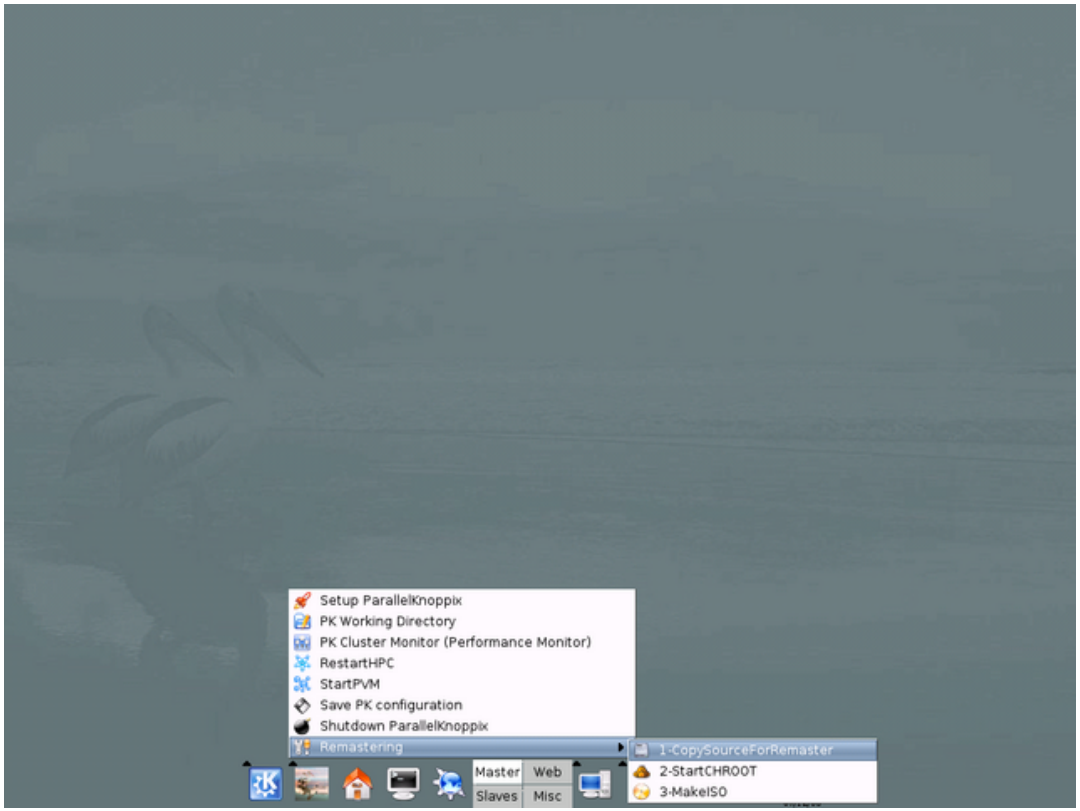
QEMU

Using QEMU is more or less similar. There's a tutorial for launching PK inside a QEMU virtual machine here. Thanks André Balsa!
Back to index

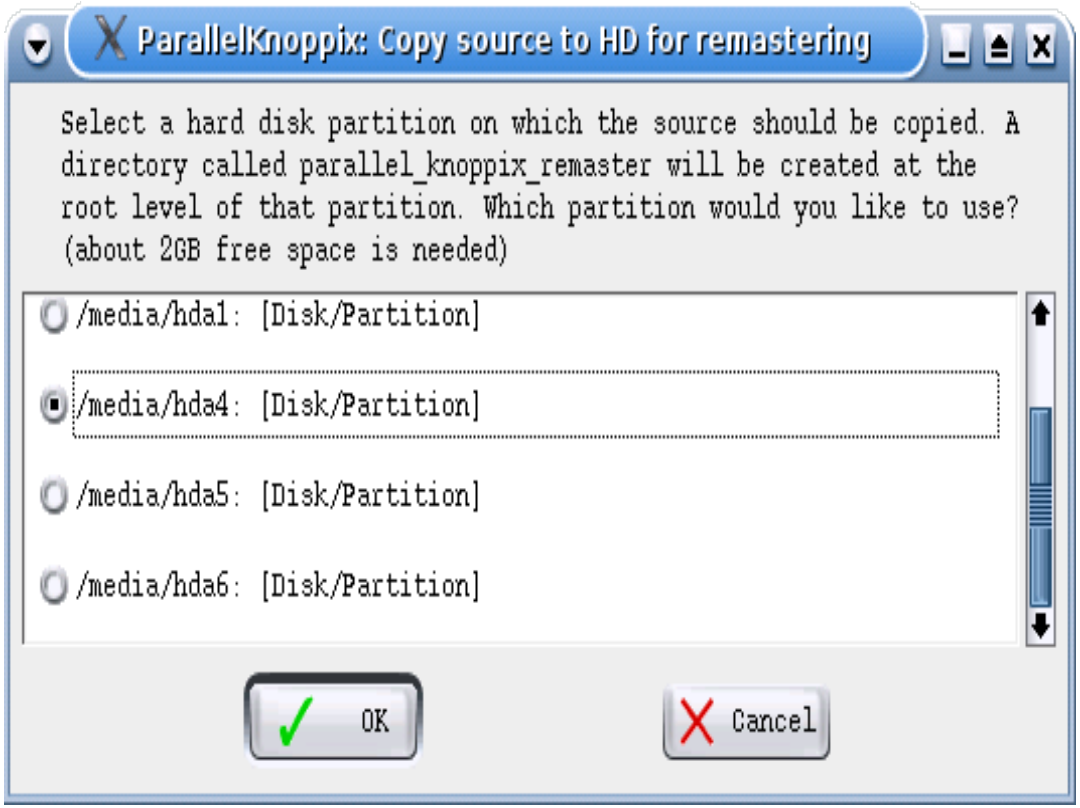
Remastering

Thanks to `aufs`, you can install new packages on the master node, using `apt-get`. But they will not be available on the compute nodes unless you repeat the process by hand on each node. You can also install new packages into `/home/knoppix`, but this requires that you save your work and reload it each time you start up. You may prefer to make your own customized version of PK - a process known as remastering. PK has some scripts that will help you to do this. Basic remastering is not *too* difficult, but you should go get your manual of swear words and make sure you know how to use them before getting started. In addition to what I explain here, you can check here and here.

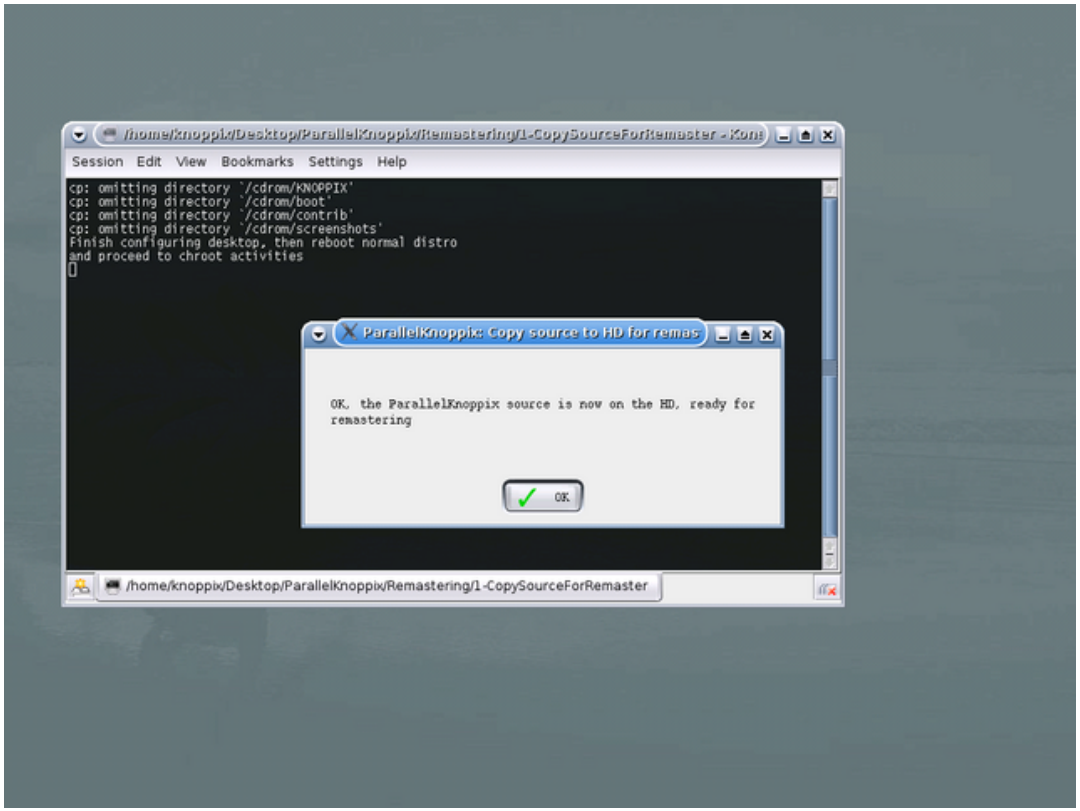
The first step is to copy the PK source to a hard disk partition, using the 1-CopySourceforRemaster script:



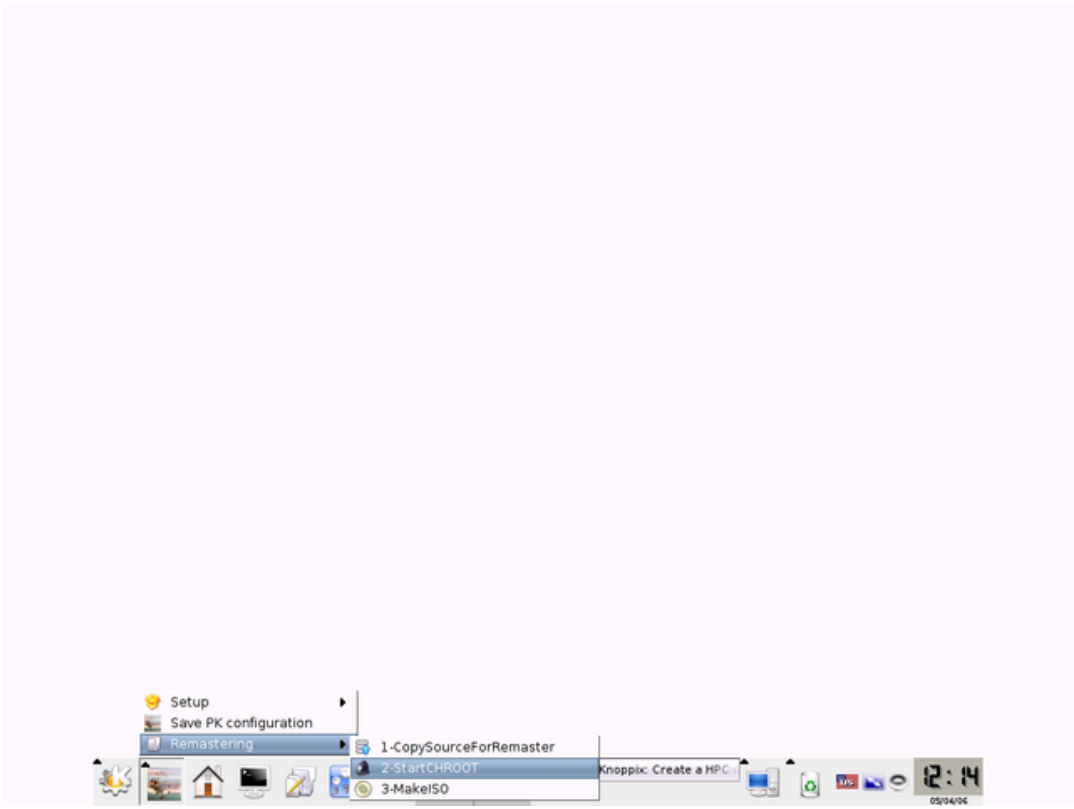
You need to choose which partition to use. You need a fair amount of space (about 4 GB should be good) since the uncompressed source (about 2 GB) as well as the CD image you will create (each one is about 650MB) need to fit there:



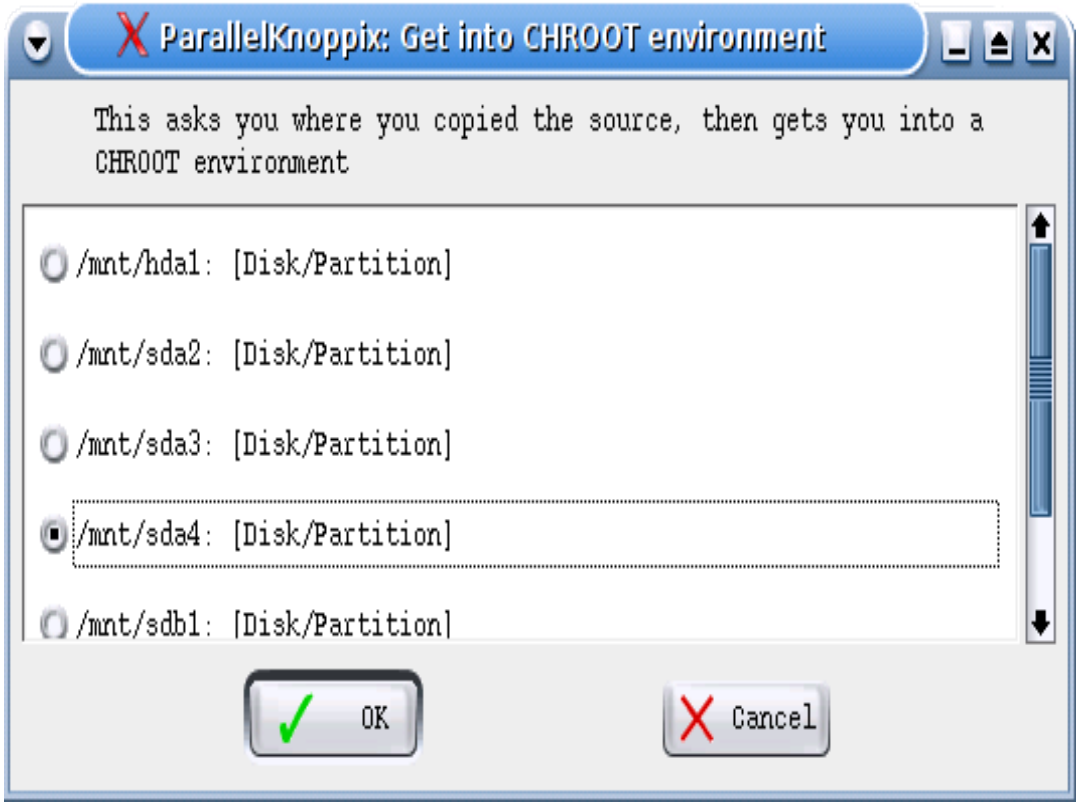
Once you select a partition, a fair amount of time goes by, as the source is copied. Eventually you see:



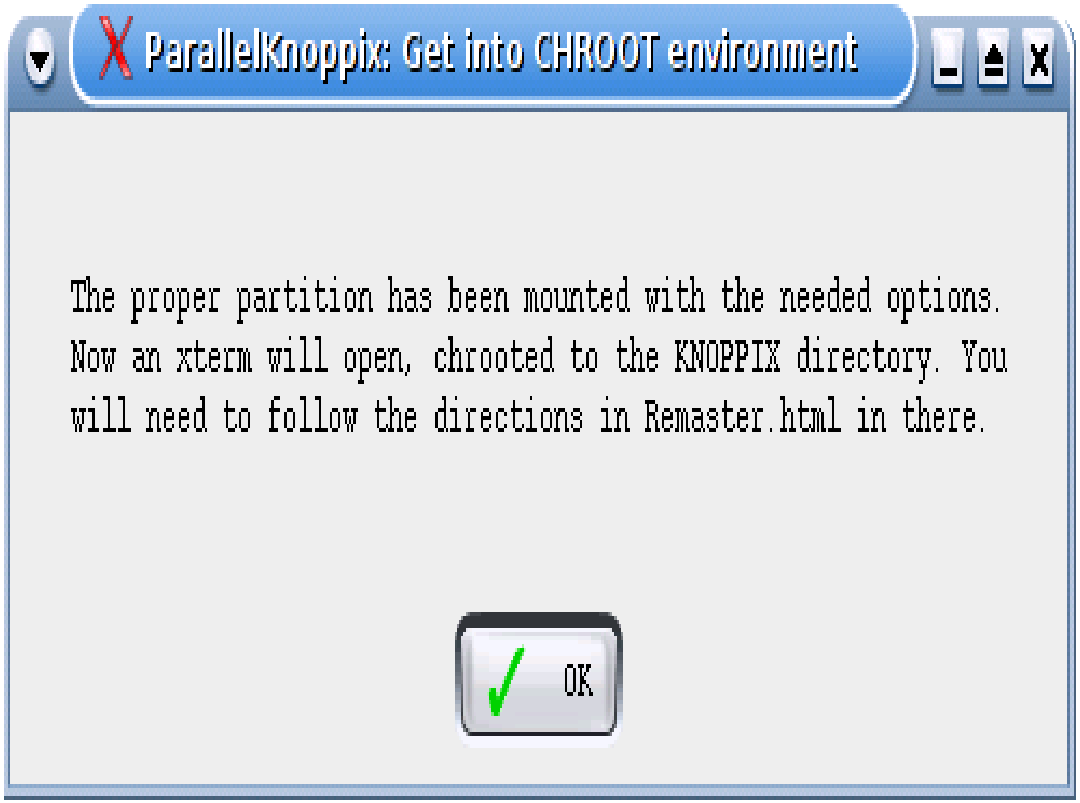
Time for step 2, entering the chroot environment as making the changes you like. Use the 2-StartCHROOT script:



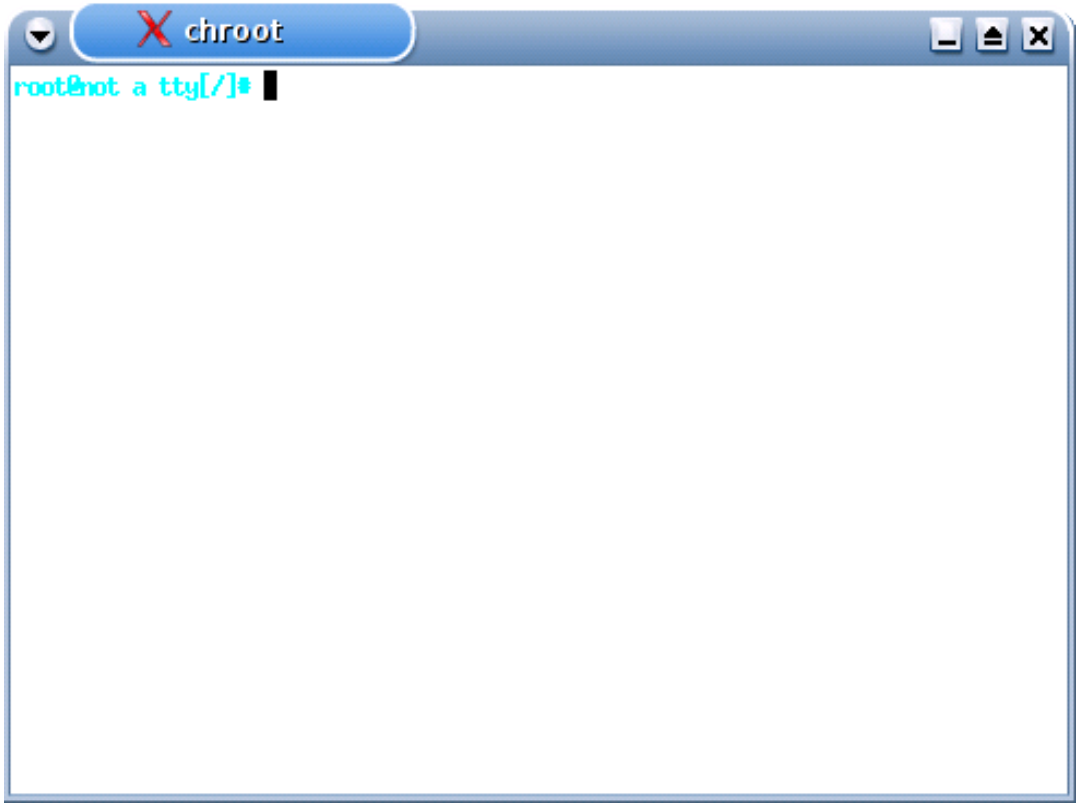
You need to tell this script where you copied the source to:



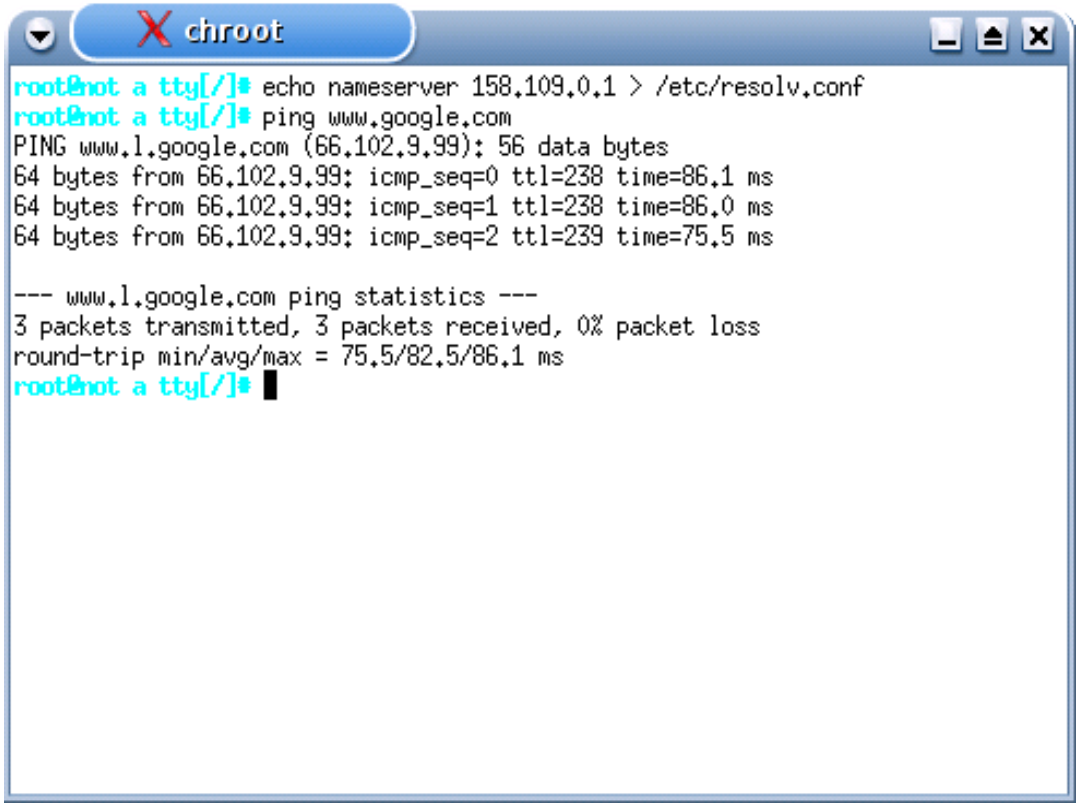
You get some information. Some (probably out of date) information is at http://pareto.uab.es/mcreel/ParallelKnoppix/Remaster.html



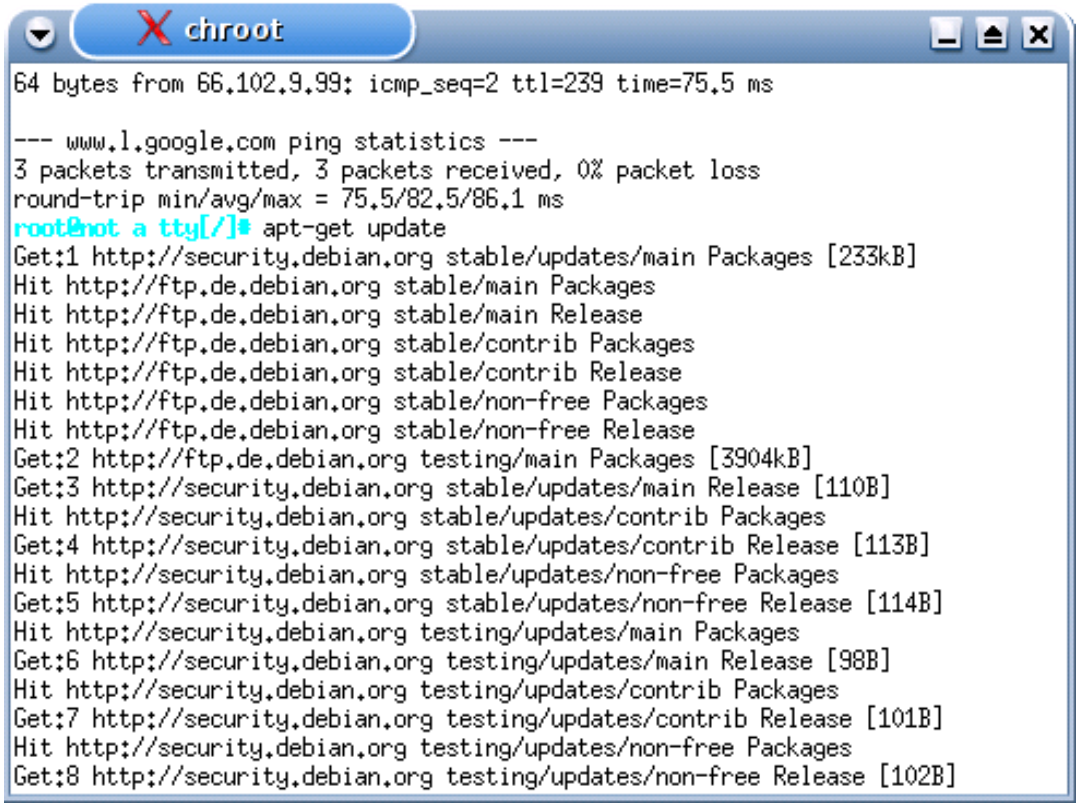
Bingo, the unusual look indicates that you are root user in a chroot environment in the PK source directory on your hard disk. If you have little experience as root user in Linux, you will probably make some big, painful mistakes, so if in doubt, type "exit" or CTRL-D. You have been warned. This is not too difficult, but you can certainly screw up your computer at this point, if you don't know what you're doing.



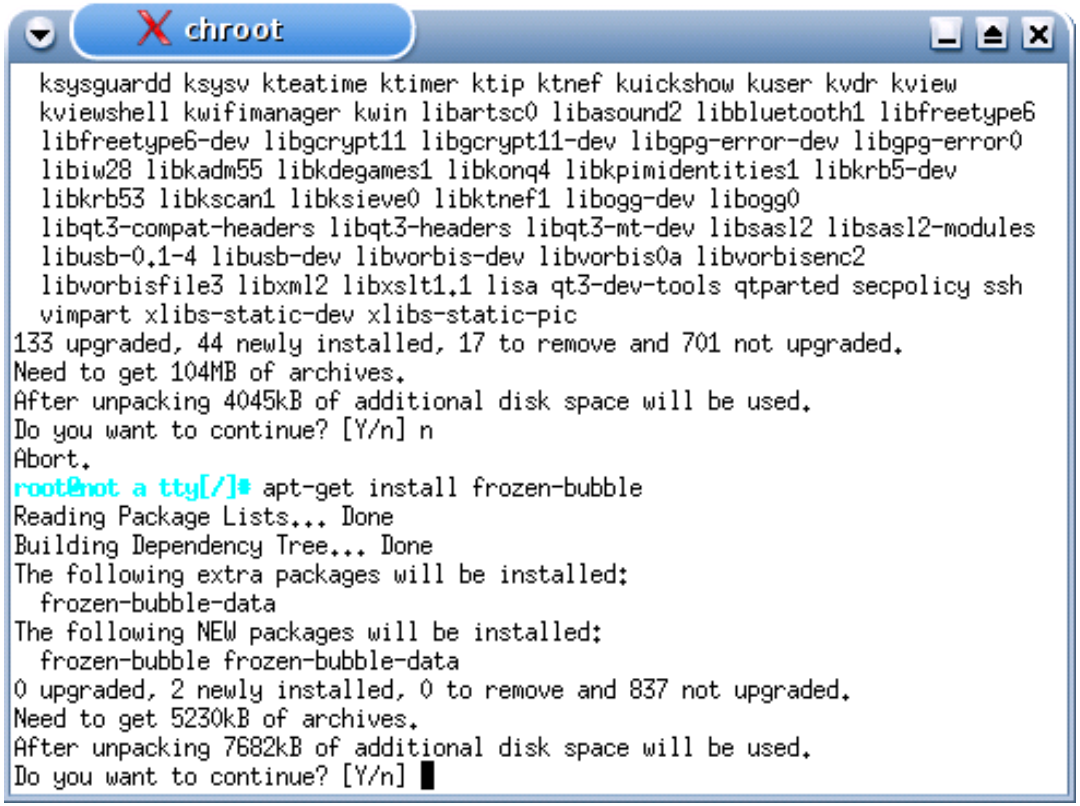
You need to get networking going. Assuming that you had it up in your ordinary PK environment (eg, boot PK while connected to a DHCP server), you should have a nameserver. You need to echo the IP of your nameserver to /etc/resolv.conf, as follows (your nameserver's IP will be different, this is just an example):



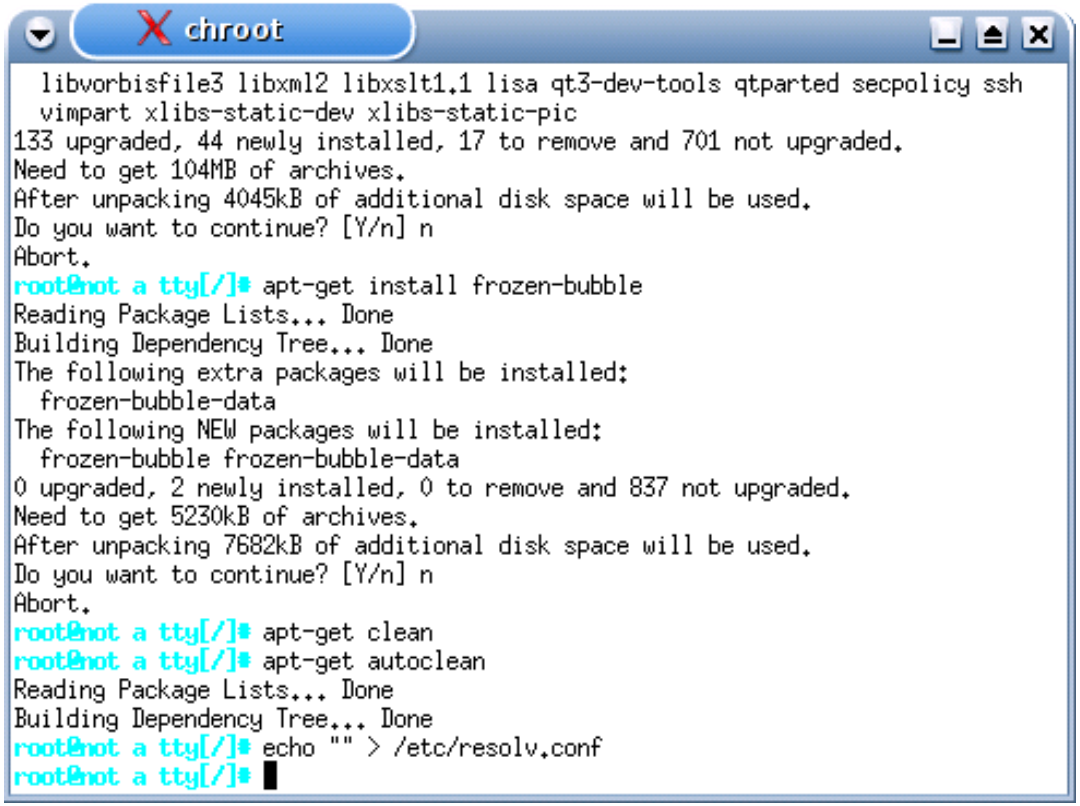
Now you're set to do the apt-get routine to your heart's content:



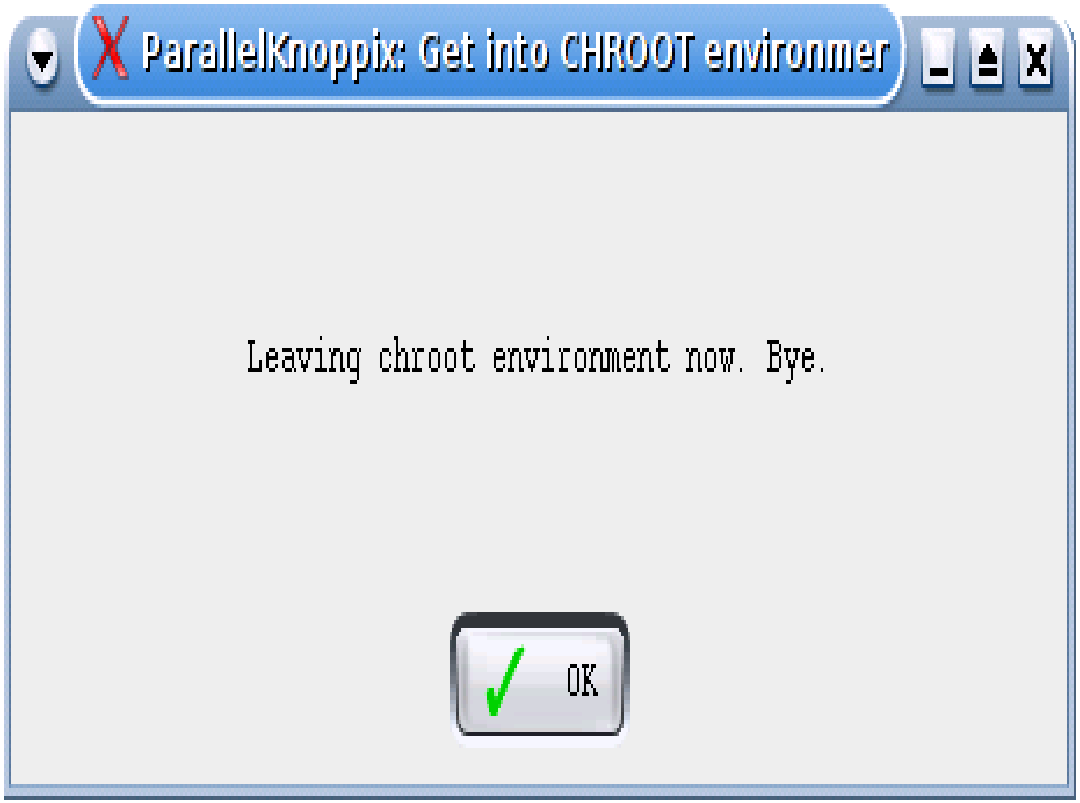
Get all that important stuff for HPC computing:



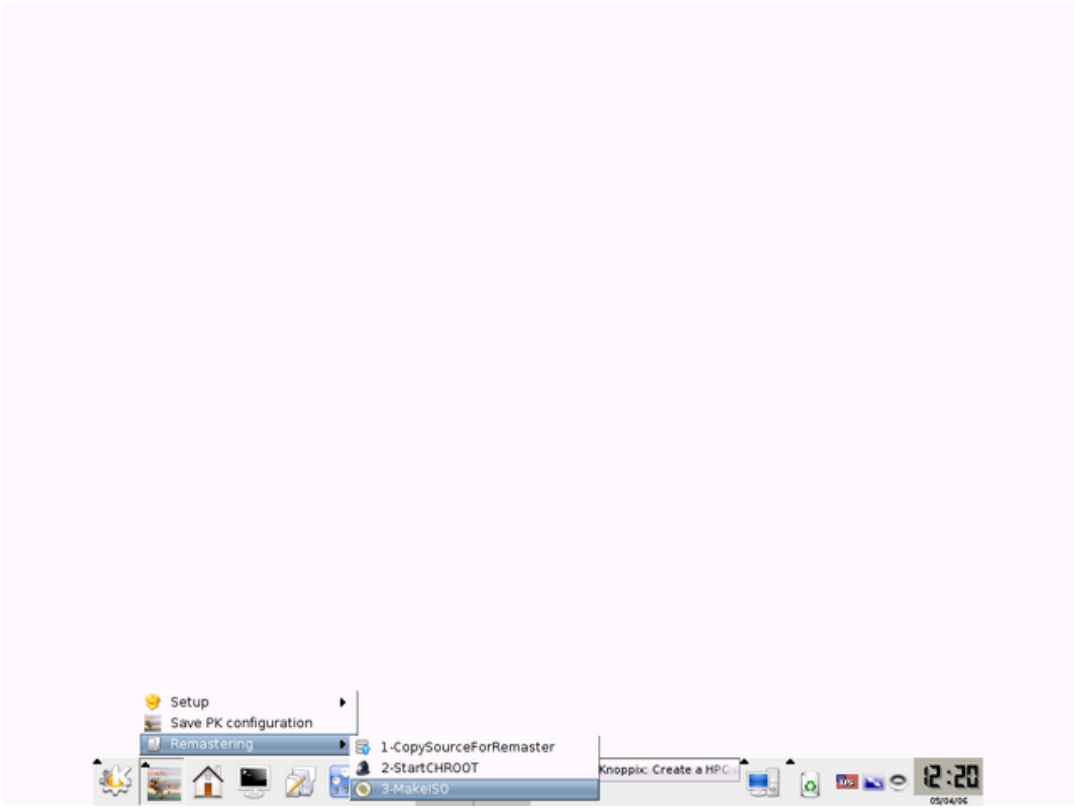
Don't overdo it, if you plan for your image to fit on a CD. Remember to clean up when you're done:



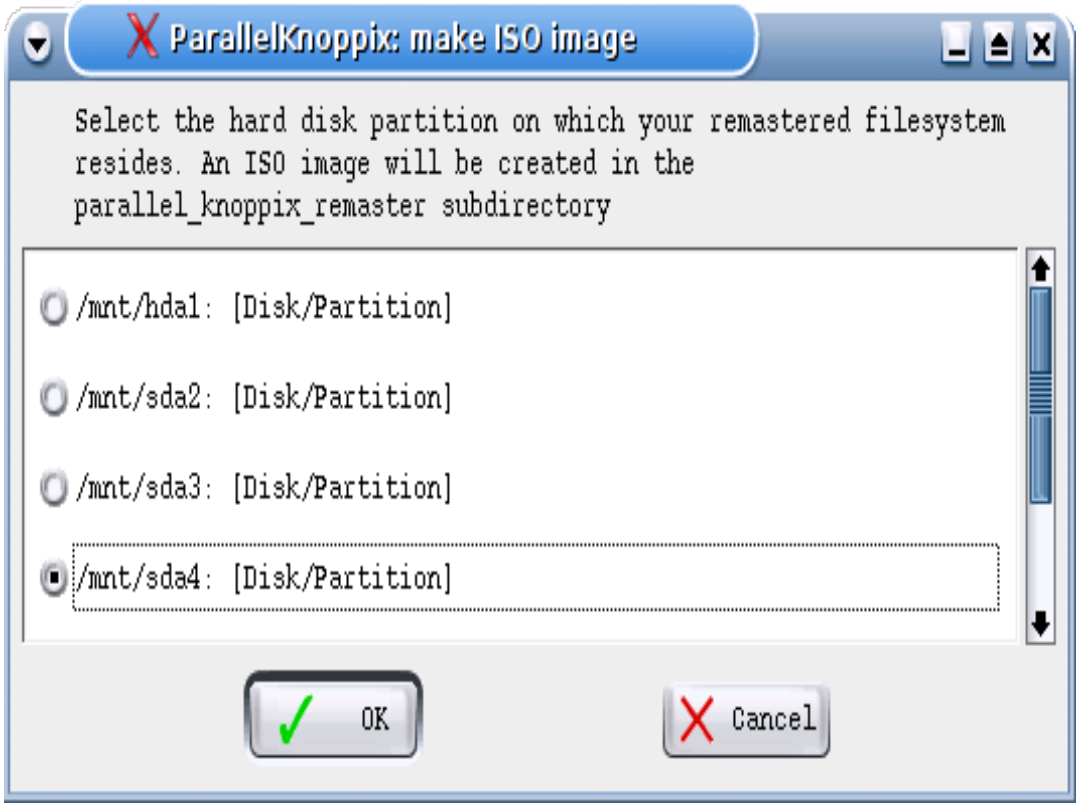
Type CTRL-D (or "exit"), and we see a friendly message:



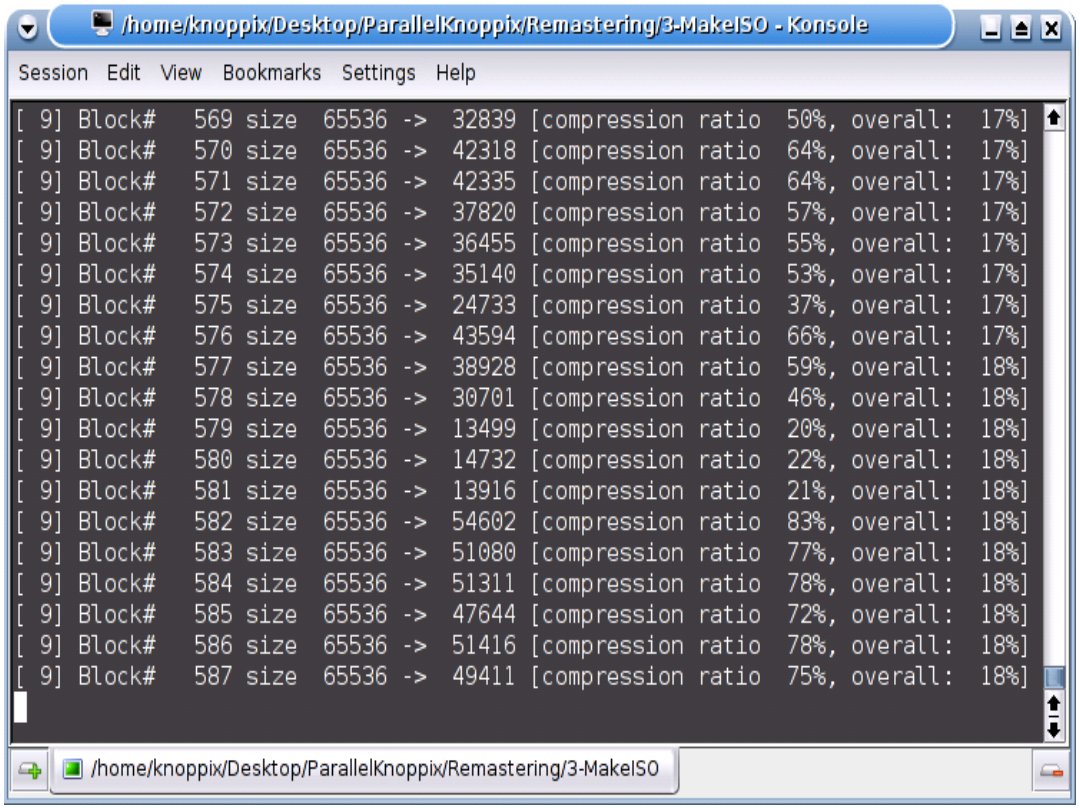
To make the ISO image, there is a 3rd script:



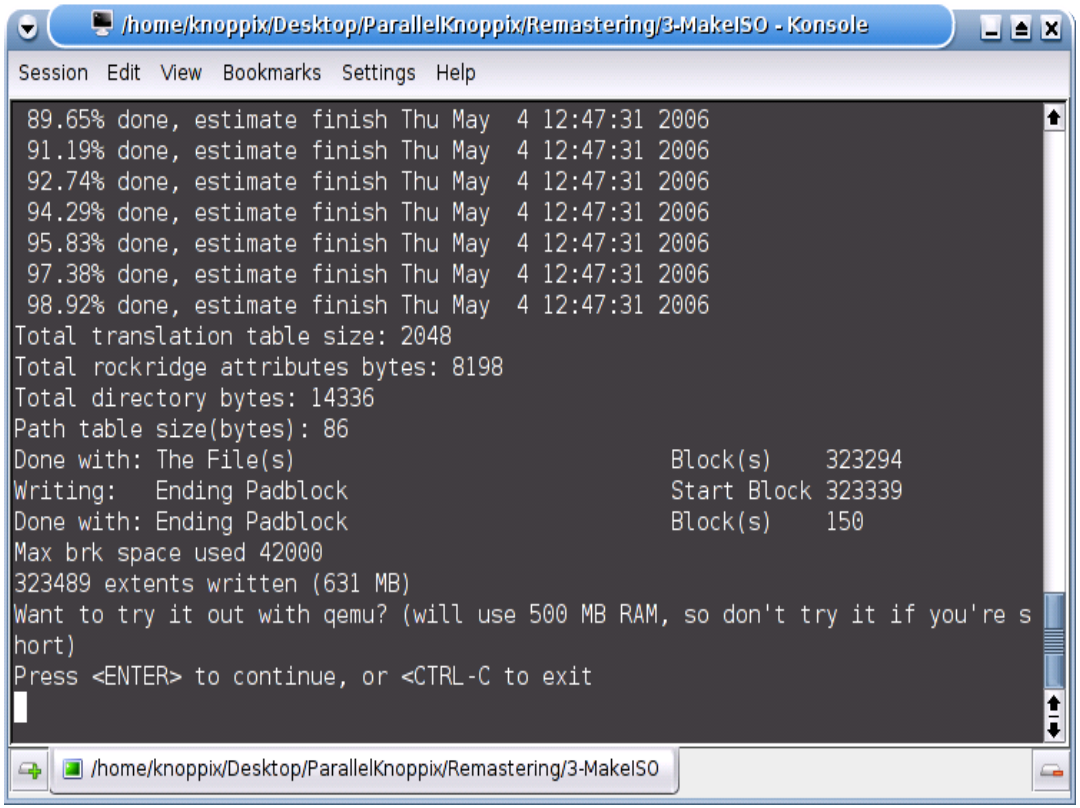
Where did you copy it to, again? (PK has a short attention span)



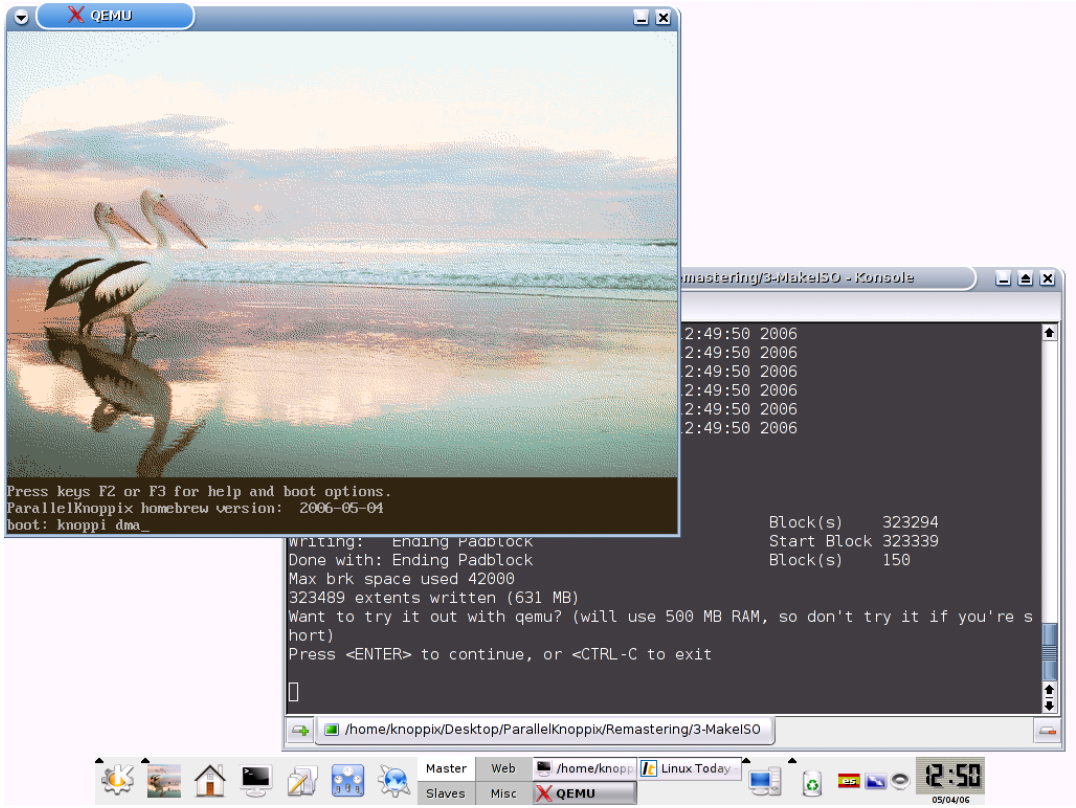
You will see a lot of information, for quite a long time:



Eventually it finishes:



If you hit <ENTER>, you get to see your remaster booting virtually, from inside the running PK. This way you can convince yourself it is working, before burning it to a CD. Amazing what you can do with free software, isn't it!



OK, that's it. Go nuts.

Back to index